

Algebraic Properties of Valued Constraint Satisfaction Problem

Marcin Kozik¹ and Joanna Ochremiak²

¹ Jagiellonian University

² University of Warsaw

Abstract. The paper presents an algebraic framework for optimization problems expressible as Valued Constraint Satisfaction Problems. Our results generalize the algebraic framework for the decision version (CSPs) provided by Bulatov et al. [SICOMP 2005].

We introduce the notions of weighted algebras and varieties and use the Galois connection due to Cohen et al. [SICOMP 2013] to link VCSP languages to weighted algebras. We show that the difficulty of VCSP depends only on the weighted variety generated by the associated weighted algebra.

Paralleling the results for CSPs we exhibit a reduction to cores and rigid cores which allows us to focus on idempotent weighted varieties. Further, we propose an analogue of the Algebraic CSP Dichotomy Conjecture; prove the hardness direction and verify that it agrees with known results for VCSPs on two-element sets [Cohen et al. 2006], finite-valued VCSPs [Thapper and Živný 2013] and conservative VCSPs [Kolmogorov and Živný 2013].

1 Introduction

An instance of the Constraint Satisfaction Problem (CSP) consists of variables (to be evaluated in a domain) and constraints restricting the evaluations. The aim is to find an evaluation satisfying all the constraints or satisfying the maximal possible number of constraints or approximating the maximal possible number of satisfied constraints etc. depending on the version of the problem. Further one can divide constraint satisfaction problems with respect to the size of the domain, the allowed constraints or the shape of the instances.

A particularly interesting version of CSP was proposed in a seminal paper of Feder and Vardi [12]. In this version a CSP is defined by a *language* which consists of a finite number of relations over a finite set. An instance of such a CSP is allowed if all the constraint relations are from this set. The goal is to determine whether an instance has a solution satisfying all the constraints.

Each language clearly defines a problem in NP; the whole family of problems is interesting for another reason: it is robust enough to include some well studied computational problems, e.g. 2-colorability, 3-SAT, solving systems of linear equations over \mathbb{Z}_p , and still is conjectured [12] not to contain problems of intermediate complexity. This conjecture is known as the Constraint Satisfaction

Dichotomy Conjecture of Feder and Vardi. Confirming this conjecture would establish CSPs as one of the largest natural subclasses of NP without problems of intermediate complexity.

The conjecture always attracted a lot of attention, but the first results, even very interesting ones, were usually very specialized (e.g. [14]). A major breakthrough appeared with a series of papers establishing *the algebraic approach to CSP* [3, 7, 16]. This deep connection with an independently developed branch of mathematics introduced a new viewpoint and provided tools necessary to tackle wide classes of CSP languages at once. At the heart of this approach lies a Galois connection between languages and clones of operations called *polymorphisms* (which completely determine the complexity of the language).

Results obtained using this new methods include a full complexity classifications for CSPs on three-element sets [5] and containing all unary relations [4, 6]. Moreover, the algebraic approach to CSP allowed to propose a boundary between the tractable and NP-complete problems: this conjecture is known as the Algebraic Dichotomy Conjecture. Unfortunately, despite many efforts (e.g. [5]), both conjectures remain open.

The Valued Constraint Satisfaction Problem (VCSP) further extends the approach proposed by Feder and Vardi. The role of constraints is played by *cost functions* describing the price of choosing particular values for variables as a part of the solution. This generalization allows to construct languages modeling standard optimization problems, for example MAX-CUT. Moreover, by allowing ∞ as a cost of a tuple, a VCSP language can additionally model every problem that CSP can model, as well as hybrid problems like MIN-VERTEX-COVER. This makes the extended framework even more general (compare the survey [17]).

A number of classes of VCSPs have been thoroughly investigated. The underlying structure suggested capturing the properties of languages of cost functions using an amalgamation of algebraic and numerical techniques [10, 25]. The first approach which provides a Galois correspondence (mirroring the Galois correspondence for CSPs) was proposed by Cohen et al. [9]. A weighted clone defined in this paper fully captures the complexity of a VCSP language.

The present paper builds on that correspondence imitating the line of research for CSPs [7]. It is organized in the following way: Section 2 contains preliminaries and basic definitions. In Section 3 we present a reduction to cores and rigid cores. Section 4 introduces a concept of a weighted algebra and a weighted variety, and shows that those notions are well behaved in the context of the Galois connection for VCSP. Reductions developed in Section 3 together with definitions from Section 4 allow us to focus on idempotent varieties. Section 5 states a conjecture postulating (for idempotent varieties) the division between the tractable and NP-hard cases of VCSP. The conjecture is clearly a strengthening of the Algebraic Dichotomy Conjecture [7]. Section 5 contains additionally the proof of the hardness direction of the conjecture as well as the reasoning showing that the conjecture agrees with complexity classifications for VCSPs on two-element sets [10], with finite-valued cost functions [25], and with conservative cost functions [19].

2 Preliminaries

2.1 The Valued Constraint Satisfaction Problem

Throughout the paper, let $\overline{\mathbb{Q}} = \mathbb{Q} \cup \{\infty\}$ denote the set of rational numbers with (positive) infinity. We assume that $x + \infty = \infty$ and $y \cdot \infty = \infty$ for $y \geq 0$. An r -ary *relation* on a set D is a subset of D^r , a *cost function* on D of arity r is a function from D^r to $\overline{\mathbb{Q}}$. We denote by Φ_D the set of all cost functions on D . A cost function which takes only finite values is called *finite-valued*. A $\{0, \infty\}$ -valued cost function is called *crisp* and can be viewed as a relation.

Definition 1. An instance of the valued constraint satisfaction problem (VCSP) is a triple $\mathcal{I} = (V, D, \mathcal{C})$ with V a finite set of variables, D a finite domain and \mathcal{C} a finite multi-set of constraints. Each constraint is a pair $C = (\sigma, \varrho)$ with σ a tuple of variables of length r and ϱ a cost function on D of arity r .

An assignment for \mathcal{I} is a mapping $s: V \rightarrow D$. The cost of an assignment s is given by $\text{Cost}_{\mathcal{I}}(s) = \sum_{(\sigma, \varrho) \in \mathcal{C}} \varrho(s(\sigma))$ (where s is applied component-wise). To solve \mathcal{I} is to find an assignment with a minimal cost, called an optimal assignment.

Example 2. In the MAX-CUT problem, one needs to find a partition of the vertices of a given graph into two sets, such that the number of edges with ends in different sets is maximal. This problem is NP-hard.

The MAX-CUT problem can be expressed as an instance of VCSP. The domain has two elements 0 and 1. Variables in the instance are vertices of the graph and for each edge e there is a constraint of a form (e, ϱ_{XOR}) , where ϱ_{XOR} is a binary cost function defined by

$$\varrho_{XOR}(x, y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$$

Any assignment of the values 0 and 1 to the variables corresponds to a partition of the graph. The cost of an assignment is equal to the number of edges of the graph minus the number of cut edges.

Any set $\Gamma \subseteq \Phi_D$ is called a *valued constraint language* over D , or simply a *language*. If all cost functions from Γ are $\{0, \infty\}$ -valued or finite-valued, we call it a *crisp* or *finite-valued* language, respectively. If Γ is a language, but not necessarily finite-valued or crisp, we sometimes stress this fact by saying that Γ is a *general-valued* language.

By $\text{VCSP}(\Gamma)$ we denote the class of all VSCP instances in which all cost functions in all constraints belong to Γ . $\text{VCSP}(\Gamma_{crisp})$, where Γ_{crisp} is the language consisting of all crisp cost functions on some fixed set D , is equivalent to the classical CSP. For an instance $\mathcal{I} \in \text{VCSP}(\Gamma)$ we denote by $\text{Opt}_{\Gamma}(\mathcal{I})$ the cost of an optimal assignment. We say that a language Γ is *tractable* if, for every finite subset $\Gamma' \subseteq \Gamma$, there exists an algorithm solving any instance $\mathcal{I} \in \text{VCSP}(\Gamma')$

in polynomial time, and we say that Γ is *NP-hard* if $\text{VCSP}(\Gamma')$ is NP-hard for some finite $\Gamma' \subseteq \Gamma$. Example 2 shows that the language $\{\varrho_{XOR}\}$ is NP-hard.

Weighted Relational Clones. We follow the exposition from [9] and define a closure operator on valued constraint languages that preserves tractability.

Definition 3. A cost function ϱ is expressible over a valued constraint language $\Gamma \subseteq \Phi_D$ if there exists an instance $\mathcal{I}_\varrho \in \text{VCSP}(\Gamma)$ and a list (v_1, \dots, v_r) of variables of \mathcal{I}_ϱ , such that

$$\varrho(x_1, \dots, x_r) = \min_{\{s: V \rightarrow D \mid s(v_i) = x_i\}} \text{Cost}_{\mathcal{I}_\varrho}(s).$$

Note that the list of variables (v_1, \dots, v_r) in the definition above might contain repeated entries. Hence, it is possible that there are no assignments s such that $s(v_i) = x_i$ for all i . We define the minimum over the empty set to be ∞ .

Definition 4. A set $\Gamma \subseteq \Phi_D$ is a weighted relational clone if it is closed under, expressibility, scaling by non-negative rational constants, and addition of rational constants. We define $\text{wRelClo}(\Gamma)$ to be the smallest weighted relational clone containing Γ .

If $\varrho(x_1, \dots, x_r) = \varrho_1(y_1, \dots, y_s) + \varrho_2(z_1, \dots, z_t)$ for some fixed choice of arguments $y_1, \dots, y_s, z_1, \dots, z_t$ from amongst x_1, \dots, x_r then the cost function ϱ is said to be obtained by *addition* from the cost functions ϱ_1 and ϱ_2 . It is easy to see that a weighted relational clone is closed under addition, and minimisation over arbitrary arguments.

The following result shows that we can restrict our attention to languages which are weighted relational clones.

Theorem 5 (Cohen et al. [9]). A valued constraint language Γ is tractable if and only if $\text{wRelClo}(\Gamma)$ is tractable, and it is NP-hard if and only if $\text{wRelClo}(\Gamma)$ is NP-hard.

Weighted polymorphisms. A k -ary operation on D is a function $f: D^k \rightarrow D$. We denote by \mathcal{O}_D the set of all finitary operations on D and by $\mathcal{O}_D^{(k)}$ the set of all k -ary operations on D . The k -ary projections, defined for all $i \in \{1, \dots, k\}$, are the operations $\pi_i^{(k)}$ such that $\pi_i^{(k)}(x_1, \dots, x_k) = x_i$. Let $f \in \mathcal{O}_D^{(k)}$ and $g_1, \dots, g_k \in \mathcal{O}_D^{(l)}$. The l -ary operation $f[g_1, \dots, g_k]$ defined by $f[g_1, \dots, g_k](x_1, \dots, x_l) = f(g_1(x_1, \dots, x_l), \dots, g_k(x_1, \dots, x_l))$ is called the *superposition* of f and g_1, \dots, g_k .

A set $C \subseteq \mathcal{O}_D$ is a *clone of operations* (or simply a *clone*) if it contains all projections on D and is closed under superposition. The set of k -ary operations in a clone C is denoted $C^{(k)}$. The smallest possible clone of operations over a fixed set D is the set of all projections on D , which we denote Π_D .

Following [9] we define a k -ary *weighting* of a clone C to be a function $\omega: C^{(k)} \rightarrow \mathbb{Q}$ such that $\sum_{f \in C^{(k)}} \omega(f) = 0$, and if $\omega(f) < 0$ then f is a projection. The set of operations to which a weighting ω assigns positive weights is called the *support* of ω and denoted $\text{supp}(\omega)$.

A new weighting of the same clone can be obtained by scaling a weighting by a non-negative rational, adding two weightings of the same arity and by the following operation called *superposition*.

Definition 6. Let ω be a k -ary weighting of a clone C and let $g_1, \dots, g_k \in C^{(l)}$. A superposition of ω and g_1, \dots, g_k is a function $\omega[g_1, \dots, g_k]: C^{(l)} \rightarrow \mathbb{Q}$ defined by

$$\omega[g_1, \dots, g_k](f') = \sum_{\{f \in C^{(k)} \mid f[g_1, \dots, g_k] = f'\}} \omega(f).$$

The sum of weights that any superposition $\omega[g_1, \dots, g_k]$ assigns to the operations in $C^{(l)}$ is equal to the sum of weights in ω , which is 0. However, it may happen that a superposition assigns a negative value to an operation that is not a projection. A superposition is said to be *proper* if the result is a valid weighting.

A non-empty set of weightings over a fixed clone C is called a *weighted clone* if it is closed under non-negative scaling, addition of weightings of equal arity and proper superposition with operations from C . For any clone of operations C , the set of all weightings over C and the set of all zero-valued weightings of C are weighted clones.

We say that an r -ary relation R on D is *compatible* with an operation $f: D^k \rightarrow D$ if, for any list of r -tuples $\mathbf{x}_1, \dots, \mathbf{x}_k \in R$ we have $f(\mathbf{x}_1, \dots, \mathbf{x}_k) \in R$ (where f is applied coordinate-wise). Let $\varrho: D^r \rightarrow \overline{\mathbb{Q}}$ be a cost function. We define $\text{Feas}(\varrho) = \{\mathbf{x} \in D^r \mid \varrho(\mathbf{x}) \text{ is finite}\}$ to be the *feasibility relation* of ϱ . We call an operation $f: D^k \rightarrow D$ a *polymorphism* of ϱ if the relation $\text{Feas}(\varrho)$ is compatible with it. For a valued constraint language Γ we denote by $\text{Pol}(\Gamma)$ the set of operations which are polymorphisms of all cost functions $\varrho \in \Gamma$. It is easy to verify that $\text{Pol}(\Gamma)$ is a clone. The set of m -ary operations in $\text{Pol}(\Gamma)$ is denoted $\text{Pol}_m(\Gamma)$.

For crisp cost functions (relations) this notion of polymorphism corresponds precisely to the standard notion of polymorphism which has played a crucial role in the complexity analysis for the CSP [3, 16].

Definition 7. Take ϱ to be a cost function of arity r on D , and let $C \subseteq \text{Pol}(\{\varrho\})$ be a clone of operations. A weighting $\omega: C^{(k)} \rightarrow \mathbb{Q}$ is called a *weighted polymorphism of ϱ* if, for any list of r -tuples $\mathbf{x}_1, \dots, \mathbf{x}_k \in \text{Feas}(\varrho)$, we have

$$\sum_{f \in C^{(k)}} \omega(f) \cdot \varrho(f(\mathbf{x}_1, \dots, \mathbf{x}_k)) \leq 0.$$

For a valued constraint language Γ we denote by $\text{wPol}(\Gamma)$ the set of those weightings of the clone $\text{Pol}(\Gamma)$ that are weighted polymorphisms of all cost functions $\varrho \in \Gamma$. The set of weightings $\text{wPol}(\Gamma)$ is a weighted clone [9].

Example 8. For any lattice-ordered set D , a function $\varrho: D^r \rightarrow \mathbb{Q}$ is called *sub-modular* if for all $\mathbf{x}_1, \mathbf{x}_2 \in D^r$

$$\varrho(\min(\mathbf{x}_1, \mathbf{x}_2)) + \varrho(\max(\mathbf{x}_1, \mathbf{x}_2)) - \varrho(\mathbf{x}_1) - \varrho(\mathbf{x}_2) \leq 0.$$

The above condition can be equivalently expressed by saying that the set of submodular functions on D is the set of cost functions with a binary weighted polymorphism ω , defined as follows:

$$\omega(f) = \begin{cases} -1 & \text{if } f \text{ is a projection,} \\ 1 & \text{if } f \text{ is one of the operations } \min \text{ or } \max, \\ 0 & \text{otherwise.} \end{cases}$$

An operation f is *idempotent* if $f(x, \dots, x) = x$. A weighted polymorphism is called *idempotent* if all operations in its support are idempotent.

An operation $f \in \mathcal{O}_D^{(k)}$ is *cyclic* if for every $x_1, \dots, x_k \in D$ we have that $f(x_1, x_2, \dots, x_k) = f(x_2, \dots, x_k, x_1)$. A weighted polymorphism is called *cyclic* if its support is non-empty and contains cyclic operations only.

A cost function ϱ is said to be *improved* by a weighting ω if ω is a weighted polymorphism of ϱ . For any set W of weightings over a fixed clone $C \subseteq \mathcal{O}_D$ we denote by $\text{Imp}(W)$ the set of cost functions on D which are improved by all weightings $\omega \in W$. The following result, together with Theorem 5, implies that tractable valued constraint languages can be characterised by weighted polymorphisms.

Theorem 9 (Cohen et al. [9]). *For any finite valued constraint language Γ , we have $\text{Imp}(\text{wPol}(\Gamma)) = \text{wRelClo}(\Gamma)$.*

For more information on the valued constraint satisfaction problem see the recent survey [17].

2.2 Algebras and varieties

In this subsection we introduce the basic concepts of universal algebra that serve us as tools later on in this paper. An *algebraic signature* is a set of function symbols together with (finite) arities. An *algebra* \mathbf{A} over a fixed signature Σ consists of a set A , called the *universe* of \mathbf{A} , and a set of *basic operations* that correspond to the symbols in the signature, i.e., if the signature contains a k -ary symbol f then the algebra has a basic operation $f^{\mathbf{A}}$, which is a function $f^{\mathbf{A}}: A^k \rightarrow A$.

A subset B of the universe of an algebra \mathbf{A} is a *subuniverse* of \mathbf{A} if it is closed under all operations of \mathbf{A} . An algebra \mathbf{B} is a *subalgebra* of \mathbf{A} if B is a subuniverse of \mathbf{A} and the operations of \mathbf{B} are restrictions of all the operations of \mathbf{A} to B . Let $(\mathbf{A}_i)_{i \in I}$ be a family of algebras (over the same signature). Their *product* $\prod_{i \in I} \mathbf{A}_i$ is an algebra with the universe equal to the cartesian product of the A_i 's and operations computed coordinate-wise. For two algebras \mathbf{A} and \mathbf{B} (over the same signature), a *homomorphism* from \mathbf{A} to \mathbf{B} is a function $h: A \rightarrow B$ that preserves all operations. It is easy to see, that an image of a homomorphism $h: A \rightarrow B$ is a subalgebra of \mathbf{B} .

Let \mathcal{K} be a class of algebras over a fixed signature Σ . We denote by $\text{S}(\mathcal{K})$ the class of all subalgebras of algebras in \mathcal{K} , by $\text{P}(\mathcal{K})$ the class of all products of

algebras in \mathcal{K} , by $P_{fin}(\mathcal{K})$ the class of all finite products, and by $H(\mathcal{K})$ the class of all homomorphic images of algebras in \mathcal{K} . If $\mathcal{K} = \{\mathbf{A}\}$ we write $S(\mathbf{A})$, $P(\mathbf{A})$, and $H(\mathbf{A})$ instead of $S(\{\mathbf{A}\})$, $P(\{\mathbf{A}\})$, and $H(\{\mathbf{A}\})$, respectively.

Similarly $\mathcal{V}(\mathcal{K})$ is the smallest class of algebras closed under all three operations. For an algebra \mathbf{A} the variety $\mathcal{V}(\{\mathbf{A}\})$ (denoted $\mathcal{V}(\mathbf{A})$) is the variety *generated* by \mathbf{A} , and $\mathcal{V}_{fin}(\mathbf{A})$ is the class of finite algebras in $\mathcal{V}(\mathbf{A})$. The variety $\mathcal{V}(\mathbf{A})$ can be characterised as follows:

Proposition 10 (Tarski [23]). *For any finite algebra \mathbf{A} , we have*

$$\mathcal{V}(\mathbf{A}) = \text{HSP}(\mathbf{A}) \quad \text{and} \quad \mathcal{V}_{fin}(\mathbf{A}) = \text{HSP}_{fin}(\mathbf{A}).$$

We say that an equivalence relation \sim on A is a *congruence* of \mathbf{A} if the following condition is satisfied for all operations f of \mathbf{A} : if for all $i \in \{1, \dots, k\}$, we have $a_i \sim b_i$, then $f(a_1, \dots, a_k) \sim f(b_1, \dots, b_k)$, where k is the arity of f . Every congruence \sim of \mathbf{A} determines a *quotient* algebra \mathbf{A}/\sim . Its universe is the set of the equivalence classes A/\sim and operations are defined using their arbitrarily chosen representatives.

A *term* t in a signature Σ is a formal expression built from variables and symbols in Σ that syntactically describes the composition of basic operations. For an algebra \mathbf{A} over Σ a *term operation* $t^{\mathbf{A}}$ is an operation obtained by composing the basic operations of \mathbf{A} according to t . Let s and t be a pair of terms in a signature Σ . We say that \mathbf{A} satisfies the *identity* $s \approx t$ if the term operations $s^{\mathbf{A}}$ and $t^{\mathbf{A}}$ are equal. We say that a class of algebras \mathcal{V} over Σ satisfies the identity $s \approx t$ if every algebra in \mathcal{V} does.

Theorem 11 (Birkhoff [2]). *A class of algebras \mathcal{V} is a variety if and only if there exists a set of identities such that \mathcal{V} contains precisely those algebras that satisfy all the identities from this set.*

It follows from Birkhoff's theorem that the variety $\mathcal{V}(\mathbf{A})$ is the class of algebras that satisfy all the identities satisfied by \mathbf{A} . Moreover, if \mathbf{A} is finite then $\mathcal{V}(\mathbf{A})$ is *locally finite*, i.e., every finitely generated algebra in $\mathcal{V}(\mathbf{A})$ is finite.

3 Core Valued Constraint Languages

For each valued constraint language Γ there is an associated algebra. It has universe D and the set of operations $\text{Pol}(\Gamma)$. If all polymorphisms of Γ are idempotent it means that the algebra $(D, \text{Pol}(\Gamma))$ satisfies the identity $f(x, \dots, x) \approx x$ for every operation f . Such algebras are called *idempotent*. In this section we prove that every finite valued constraint language has a computationally equivalent valued constraint language whose associated algebra is idempotent.

3.1 Positive Clone.

Those polymorphisms of a given language Γ which are assigned a positive weight by some weighted polymorphisms $\omega \in \text{wPol}(\Gamma)$ are of special interest in the rest of the paper. We begin this section by proving that they form a clone.

Let \mathcal{C} be a weighted clone over a set D . The following proposition shows that the set $\bigcup_{\omega \in \mathcal{C}} \text{supp}(\omega)$, together with the set of projections Π_D , is a clone. We call it the *positive clone* of \mathcal{C} and denote by C^+ (if \mathcal{C} is $\text{wPol}(F)$ then C^+ is denoted by $\text{Pol}^+(F)$).

Proposition 12. *If \mathcal{C} is a weighted clone then C^+ is a clone.*

We will use the following technical lemma (Lemma 6.5 from [9]). It implies that any weighting that can be expressed as a weighted sum of arbitrary superpositions can also be expressed as a superposition of a weighted sum of proper superpositions.

Lemma 13. *Let \mathcal{C} be a weighted clone, and let ω_1 and ω_2 be weightings in \mathcal{C} , of arity k and l respectively. For any m -ary operations $f_1, \dots, f_k, g_1, \dots, g_l$ of \mathcal{C} :*

$$c_1\omega_1[f_1, \dots, f_k] + c_2\omega_2[g_1, \dots, g_l] = \omega[f_1, \dots, f_k, g_1, \dots, g_l],$$

where

$$\omega = c_1\omega_1[\pi_1^{(k+l)}, \dots, \pi_k^{(k+l)}] + c_2\omega_2[\pi_{k+1}^{(k+l)}, \dots, \pi_{k+l}^{(k+l)}].$$

Proof. We need to show that the set C^+ is closed under superposition. Take a k -ary operation f and a list of l -ary operations g_1, \dots, g_k that all belong to C^+ .

If f is a projection there is nothing to prove. Otherwise there is a weighting $\omega \in \mathcal{C}$ such that $\omega(f) > 0$. Similarly for each g_i which is not a projection we find ω_i such that $\omega_i(g_i) > 0$ (if g_i is a projection we put ω_i to be the zero-valued l -ary weighting).

Now, there exist non-negative rational numbers i_j such that the sum

$$\omega[g_1, \dots, g_k] + i_1\omega_1[\pi_1^l, \dots, \pi_l^l] + \dots + i_k\omega_k[\pi_1^l, \dots, \pi_l^l]$$

is a valid weighting. By Lemma 13 this weighting can be obtained as a superposition of a sum of proper superpositions and therefore belongs to \mathcal{C} which finishes the proof.

3.2 Cores.

Let Γ be a valued constraint language with a domain D . For $S \subseteq D$ we denote by $\Gamma[S]$ the valued constraint language defined on a domain S and containing the restriction of every cost function $\varrho \in \Gamma$ to S . We show that Γ has a computationally equivalent valued constraint language Γ' such that $\text{Pol}_1^+(\Gamma')$ contains only bijective operations. Such a language is called a *core*. Moreover, Γ' can be chosen to be equal $\Gamma[S]$ for some $S \subseteq D$.

Proposition 14. *For every valued constraint language Γ there exists a core language Γ' , such that the valued constraint language Γ is tractable if and only if Γ' is tractable, and it is NP-hard if and only if Γ' is NP-hard.*

We prove the above result by generalizing the arguments for finite-valued languages given in [15, 25]. We need an auxiliary lemma.

Lemma 15. *For a valued constraint language Γ , let $f \in \text{Pol}_1^+(\Gamma)$ and let $\mathcal{I} \in \text{VCSP}(\Gamma)$. If s is an optimal assignment for \mathcal{I} , then $f(s)$ is also optimal.*

Proof. Let f , \mathcal{I} and s be like in the statement of the lemma. Observe that $\text{Cost}_{\mathcal{I}}$ (see Definition 1) can be seen as a cost function whose arity is equal to the number of variables in \mathcal{I} . Moreover, $\text{Cost}_{\mathcal{I}}$ belongs to $\text{wRelClo}(\Gamma)$ as it is clearly expressible over Γ . If $\text{Cost}_{\mathcal{I}}(s) = \infty$ then there is no assignment with a finite cost and we are done.

Assume that $\text{Cost}_{\mathcal{I}}(s) < \infty$, which means that $s \in \text{Feas}(\text{Cost}_{\mathcal{I}})$. If $f \neq \text{id}$, then there exists a weighted polymorphism ω with $\omega(f) > 0$. By definition the following inequality is satisfied:

$$\sum_{g \in \text{Pol}_1(\Gamma)} \omega(g) \cdot \text{Cost}_{\mathcal{I}}(g(s)) \leq 0.$$

Without loss of generality we can assume that $\omega(\text{id}) = -1$. Then we have that $\sum_{g \in \text{supp}(\omega)} \omega(g) = 1$ and the inequality above can be rewritten as

$$\sum_{g \in \text{supp}(\omega)} \omega(g) \cdot \text{Cost}_{\mathcal{I}}(g(s)) \leq \text{Cost}_{\mathcal{I}}(s).$$

On the other hand,

$$\sum_{g \in \text{supp}(\omega)} \omega(g) \cdot \text{Cost}_{\mathcal{I}}(g(s)) \geq \sum_{g \in \text{supp}(\omega)} \omega(g) \cdot \text{Cost}_{\mathcal{I}}(s) = \text{Cost}_{\mathcal{I}}(s).$$

Therefore $\text{Cost}_{\mathcal{I}}(g(s)) = \text{Cost}_{\mathcal{I}}(s)$ for each operation $g \in \text{supp}(\omega)$. Since $f \in \text{supp}(\omega)$ and s is optimal, $f(s)$ is also optimal.

Proof. (of Proposition 14) Let Γ be a valued constraint language over a domain D . Suppose that there is a unary polymorphism $f \in \text{Pol}^+(\Gamma)$ that is not bijective. Let $\Gamma' = \Gamma[f(D)]$, where $f(D) \subsetneq D$ denotes the range of f . There is a natural correspondence between instances of $\text{VCSP}(\Gamma')$ and instances of $\text{VCSP}(\Gamma)$, induced by the correspondence between functions in Γ and their restrictions in Γ' . For any instance \mathcal{I}' of $\text{VCSP}(\Gamma')$ the corresponding instance \mathcal{I} of $\text{VCSP}(\Gamma)$ has the same variables. The cost function ϱ' in each constraint is replaced by any cost function ϱ from Γ , which is equal to ϱ' when restricted to $f(D)$. We show that $\text{Opt}_{\Gamma}(\mathcal{I}) = \text{Opt}_{\Gamma'}(\mathcal{I}')$.

Any assignment for \mathcal{I}' is also an assignment for \mathcal{I} , and hence $\text{Opt}_{\Gamma}(\mathcal{I}) \leq \text{Opt}_{\Gamma'}(\mathcal{I}')$. Furthermore, by Lemma 15 for each s that is an optimal assignment for \mathcal{I} , we have

$$\text{Cost}_{\mathcal{I}}(s) = \text{Cost}_{\mathcal{I}}(f(s)) = \text{Cost}_{\mathcal{I}'}(f(s)).$$

Therefore, $\text{Opt}_{\Gamma}(\mathcal{I}) \geq \text{Opt}_{\Gamma'}(\mathcal{I}')$.

It follows that $\text{VCSP}(\Gamma)$ is tractable if and only if $\text{VCSP}(\Gamma')$ is tractable, and it is NP-hard if and only if $\text{VCSP}(\Gamma')$ is NP-hard. Moreover, the valued constraint language Γ' is defined over a smaller domain. We replace Γ with Γ' and repeat this procedure, until we obtain a language Γ' that is a core.

For core languages we characterize the set of unary weighted polymorphisms.

Proposition 16. *Let Γ be a core valued constraint language. A unary weighting ω is a weighted polymorphism of Γ if and only if it assigns positive weights only to such bijective operations $f \in \text{Pol}_1(\Gamma)$ that, for all cost functions $\varrho \in \Gamma$, satisfy $\varrho \circ f = \varrho$.*

Proof. If a valid unary weighting ω assigns positive weights only to such operations $f \in \text{Pol}_1(\Gamma)$ that, for all cost functions $\varrho \in \Gamma$, satisfy $\varrho \circ f = \varrho$, then for each $\varrho \in \Gamma$ and a tuple $\mathbf{x} \in \text{Feas}(\varrho)$

$$\sum_{f \in \text{Pol}_1(\Gamma)} \omega(f) \cdot \varrho(f(\mathbf{x})) = \sum_{f \in \text{Pol}_1(\Gamma)} \omega(f) \cdot \varrho(\mathbf{x}) = 0,$$

and ω is clearly a weighted polymorphism of Γ .

For the other direction, let ω be a unary weighted polymorphism of Γ , such that $\text{supp}(\omega) \neq \emptyset$. Without loss of generality assume that $\omega(\text{id}) = -1$. Since Γ is a core language, the operations $g \in \text{supp}(\omega)$ are bijective. For $\varrho \in \Gamma$ and a tuple $\mathbf{x} \in \text{Feas}(\varrho)$ for which ϱ takes the minimal value, we have

$$\begin{aligned} \sum_{g \in \text{supp}(\omega)} \omega(g) \cdot \varrho(g(\mathbf{x})) + \omega(\text{id}) \cdot \varrho(\mathbf{x}) &\leq 0, \text{ hence} \\ \varrho(\mathbf{x}) &\geq \sum_{g \in \text{supp}(\omega)} \omega(g) \cdot \varrho(g(\mathbf{x})) \geq \sum_{g \in \text{supp}(\omega)} \omega(g) \cdot \varrho(\mathbf{x}) = \varrho(\mathbf{x}). \end{aligned}$$

Therefore $\varrho(g(\mathbf{x})) = \varrho(\mathbf{x})$ for each $g \in \text{supp}(\omega)$, which means that the operations in the support preserve the minimal weight.

Note that, since each $g \in \text{supp}(\omega)$ is bijective, it determines a bijection of the set $\text{Feas}(\varrho)$. We have shown that this bijection preserves the set of tuples with minimal weight. It can be similarly shown by induction that it preserves the set of tuples with any other fixed weight. Hence, we have proved that $\varrho \circ g = \varrho$ for all $g \in \text{supp}(\omega)$.

This implies that, for any core language Γ , a unary polymorphism belongs to $\text{Pol}^+(\Gamma)$ if and only if it is bijective and preserves all cost functions in Γ .

Let Γ be a finite core valued constraint language over a domain D . For each arity m we fix an enumeration of all the elements of D^m . This allows us to treat every m -ary operation $f \in \mathcal{O}_D^{(m)}$ as a $|D^m|$ -tuple. We define a $|D^m|$ -ary cost function in $\text{wRelClo}(\Gamma)$ that precisely distinguishes the m -ary operations in the positive clone from all the other m -ary polymorphisms. To do this we need the following technical lemma, which is a variant of the well known Farkas' Lemma used in linear programming:

Lemma 17 (Farkas [11]). *Let S and T be finite sets of indices, where T is a disjoint union of two subsets, T_{\geq} and $T_{=}$. For all $i \in S$, and all $j \in T$, let $a_{i,j}$ and b_j be rational numbers. Exactly one of the following holds:*

- Either there exists a set of non-negative rational numbers $\{z_i \mid i \in S\}$ and a rational number C such that

$$\text{for each } j \in R_{\geq}, \quad \sum_{i \in S} a_{i,j} z_i \geq b_j + C,$$

$$\text{for each } j \in R_{=}, \quad \sum_{i \in S} a_{i,j} z_i = b_j + C.$$

- Or else there exists a set of rational numbers $\{y_j \mid j \in T\}$ such that $\sum_{j \in T} y_j = 0$ and

$$\text{for each } j \in T_{\geq}, \quad y_j \geq 0,$$

$$\text{for each } i \in S, \quad \sum_{j \in T} y_j a_{i,j} \leq 0,$$

$$\text{and } \sum_{j \in T} y_j b_j > 0.$$

The set $\{y_j \mid j \in T\}$ defined in the lemma is called a *certificate of unsolvability*.

Proposition 18. *Let Γ be a finite core valued constraint language over a domain D . For every m there exists a cost function $\varrho: \mathcal{O}_D^{(m)} \rightarrow \overline{\mathbb{Q}}$ in $\text{wRelClo}(\Gamma)$, and a rational number P , such that for every $f \in \mathcal{O}_D^{(m)}$ the following conditions are satisfied:*

1. $\varrho(f) \geq P$,
2. $\varrho(f) < \infty$ if and only if $f \in \text{Pol}(\Gamma)$,
3. $\varrho(f) = P$ if and only if $f \in \text{Pol}^+(\Gamma)$.

Proof. The cost function ϱ is given by a sum of all cost functions in Γ with positive coefficients that we define later on.

Like in the classical CSP, a cost function whose feasibility relation contains exactly those $|D^m|$ -tuples which are m -ary polymorphisms of Γ is defined by:

$$\sum_{\substack{\varrho \in \Gamma \\ (\mathbf{a}_1, \dots, \mathbf{a}_m) \in (\text{Feas}(\varrho))^m}} \varrho(x_{\mathbf{b}_1}, \dots, x_{\mathbf{b}_r}),$$

where $\mathbf{b}_i(j) = \mathbf{a}_j(i)$, and r is the arity of ϱ . For each summand we introduce a variable $z_{\varrho, \mathbf{a}_1, \dots, \mathbf{a}_m}$ and, for each $f \in \text{Pol}_m^+(\Gamma)$ we write:

$$\sum_{\substack{\varrho \in \Gamma \\ (\mathbf{a}_1, \dots, \mathbf{a}_m) \in (\text{Feas}(\varrho))^m}} z_{\varrho, \mathbf{a}_1, \dots, \mathbf{a}_m} \varrho(f(\mathbf{b}_1), \dots, f(\mathbf{b}_r)) = 0 + C,$$

while for each $f \in \text{Pol}_m(\Gamma) \setminus \text{Pol}_m^+(\Gamma)$:

$$\sum_{\substack{\varrho \in \Gamma \\ (\mathbf{a}_1, \dots, \mathbf{a}_m) \in (\text{Feas}(\varrho))^m}} z_{\varrho, \mathbf{a}_1, \dots, \mathbf{a}_m} \varrho(f(\mathbf{b}_1), \dots, f(\mathbf{b}_r)) \geq 1 + C,$$

where $\mathbf{b}_i(j) = \mathbf{a}_j(i)$, and r is the arity of ρ .

By putting the above equalities and inequalities together we obtain a system of linear inequalities and equations. By Lemma 17 there are two mutually exclusive possibilities. First, there may exist a set of non-negative rational numbers $z_{\rho, \mathbf{a}_1, \dots, \mathbf{a}_m}$ and a rational number C , such that this system is satisfied. Then the proposition is proved: items 1. and 3. follow trivially from construction. Item 2. follows by definition of the cost function.

Otherwise, there exists a set $\{y_f \mid f \in \text{Pol}_m(\Gamma)\}$ which forms the certificate of unsolvability. Then let us consider a weighting defined by $\omega(f) = y_f$. If ω is a valid weighting, then it is an m -ary weighted polymorphism of Γ . Moreover, ω assigns to all operations in $\text{Pol}_m(\Gamma) \setminus \text{Pol}_m^+(\Gamma)$ non-negative weights that sum up to a positive number. Hence, for some $h \in \text{Pol}_m(\Gamma) \setminus \text{Pol}_m^+(\Gamma)$, we have $\omega(h) > 0$, which contradicts $h \notin \text{Pol}_m^+(\Gamma)$. If it happens that $y_g < 0$ for some operation $g \in \text{Pol}_m^+(\Gamma)$ that is not a projection, then there exists an m -ary weighted polymorphism of Γ which assigns a positive weight to g . By scaling it and adding to ω (as in the proof of Proposition 12), we obtain the weighted polymorphism needed for the contradiction.

3.3 Rigid cores.

We further reduce the class of languages that we need to consider. Let Γ be a core valued constraint language over an n -element domain $D = \{d_1, \dots, d_n\}$. For each $i \in \{1, \dots, n\}$, let

$$N_i(x) = \begin{cases} 0 & \text{if } x = d_i, \\ \infty & \text{otherwise.} \end{cases}$$

and let Γ_c denote the valued constraint language obtained from Γ by adding all cost functions N_i . Observe that $\text{Pol}(\Gamma_c) = \text{IdPol}(\Gamma)$, where by $\text{IdPol}(\Gamma)$ we denote the set of idempotent polymorphisms of the language Γ . Hence, the only unary polymorphism of Γ_c is the identity, which also means that there is only one unary weighted polymorphism of Γ_c – the zero-valued polymorphism.

Definition 19. *A valued constraint language Γ is a rigid core if there is exactly one unary polymorphism of Γ , which is the identity.*

The notion of rigid core corresponds to the classical notion of rigid core considered in CSP [7]. A valued constraint language Γ is a rigid core if the set of feasibility relations of all cost functions from Γ is a rigid core in the standard sense, which is also equivalent to all polymorphisms of Γ being idempotent.

We now prove a result which, together with Proposition 14, implies that for each finite language Γ , there is a computationally equivalent language that is a rigid core.

Proposition 20. *Let Γ be a valued constraint language which is finite and a core. The valued constraint language Γ_c is a rigid core. Moreover, Γ is tractable if and only if Γ_c is tractable, and Γ is NP-hard if and only if Γ_c is NP-hard.*

Proof. Let Γ be a finite core valued constraint language over a domain $D = \{d_1, \dots, d_n\}$. It follows from Proposition 18 that there exist an n -ary cost function $N \in \text{wRelClo}(\Gamma)$, and positive rational numbers $P < Q$, such that the following conditions are satisfied:

- $N(x_1, \dots, x_n) = P$ if and only if the unary operation g defined by $d_i \mapsto x_i$ belongs to $\text{Pol}^+(\Gamma)$,
- $N(x_1, \dots, x_n) > Q$ if and only if the unary operation g defined by $d_i \mapsto x_i$ belongs to $\text{Pol}(\Gamma) \setminus \text{Pol}^+(\Gamma)$,
- otherwise $N(x_1, \dots, x_n) = \infty$.

Assume without loss of generality that $N \in \Gamma$. We show a polynomial-time Turing reduction from $\text{VCSP}(\Gamma_c)$ to $\text{VCSP}(\Gamma)$.

Let $\mathcal{I}_c = (V_c, D, \mathcal{C}_c)$ be an instance of $\text{VCSP}(\Gamma_c)$. The set of variables V in the new instance \mathcal{I} is a disjoint union of V_c and $\{v_1, \dots, v_n\}$. For every constraint of the form $((v), N_i)$ in \mathcal{C}_c we:

- add a constraint $((v, v_i), \varrho_{=})$, where

$$\varrho_{=}(x, y) = \begin{cases} 0 & \text{if } x = y, \\ \infty & \text{otherwise} \end{cases}$$

(this cost function is expressible over every valued constraint language, so without loss of generality we can assume that $\varrho_{=} \in \Gamma$),

- remove the constraint $((v), N_i)$ from \mathcal{C}_c .

We obtain a new set of constraints \mathcal{C}_1 , where all cost functions are already from Γ .

Let C be the sum of weights that all cost functions in all constraints in \mathcal{C}_1 assign to all tuples in their feasibility relations. The final set of constraints \mathcal{C} additionally contains m constraints of the form $((v_1, \dots, v_n), N)$, where m is big enough to ensure that $m \cdot (Q - P) > C$.

There are three possibilities:

- If $\text{Opt}_{\Gamma}(\mathcal{I}) = \infty$ then no assignment for \mathcal{I}_c has a finite cost. Suppose otherwise and let s_c be an assignment for \mathcal{I}_c with a finite cost. Then s_c gives rise to an assignment s for \mathcal{I} with a finite cost. It coincides with s_c on V_c and for each $i \in \{1, \dots, n\}$, we set $s(v_i) = d_i$.
- The optimal assignment s for \mathcal{I} satisfies $N(s(v_1, \dots, v_n)) = P$. Then the tuple $s(v_1, \dots, v_n)$ determines a unary operation g , defined by $d_i \mapsto s(v_i)$. The operation g , by the definition of the cost function N , belongs to the positive clone $\text{Pol}^+(\Gamma)$. Hence, g^{-1} also belongs to the positive clone. Since Γ is a core, the assignment $g^{-1}(s)$ is optimal for \mathcal{I} . Its restriction onto V_c is an optimal assignment for \mathcal{I}_c .
- The optimal assignment s for \mathcal{I} satisfies $N(s(v_1, \dots, v_n)) > Q$. While there are m constraints of the form $((v_1, \dots, v_n), N)$, we have

$$\text{Cost}_{\mathcal{I}}(s) \geq m \cdot Q > m \cdot P + C.$$

If there was any assignment s_c for \mathcal{I}_c with a finite cost, the corresponding assignment s for \mathcal{I} would satisfy $Cost_{\mathcal{I}}(s) < m \cdot P + C$, which gives a contradiction, and implies that $Opt_{\Gamma_c}(\mathcal{I}_c) = \infty$.

If Γ is a core language then the positive clone of Γ_c contains precisely the idempotent operations from the positive clone of Γ . To show this, we first prove the following lemma:

Lemma 21. *Let Γ be a core valued constraint language. For every weighted polymorphism $\omega \in \text{wPol}(\Gamma)$ there exists an idempotent weighted polymorphism $\omega' \in \text{wPol}(\Gamma)$ such that $\text{supp}(\omega) \cap \text{IdPol}(\Gamma) \subseteq \text{supp}(\omega')$. Moreover, if ω is cyclic then ω' can be chosen to be cyclic.*

Proof. Consider a weighted polymorphism $\omega \in \text{wPol}(\Gamma)$. Take a non-idempotent operation $g \in \text{supp}(\omega)$ and let h be a unary operation defined by $h(x) = g(x, \dots, x)$. Since $\text{Pol}^+(\Gamma)$ is a clone of operations, $h \in \text{Pol}^+(\Gamma)$. Then by Proposition 16 the operation h is bijective and preserves all cost functions in Γ . We modify the weighted polymorphism ω by adding $\omega(g)$ to the weight of the idempotent operation $h^{-1} \circ g$ and then assigning weight 0 to the operation g . It is straightforward to check that the new weighting is a weighted polymorphism of Γ . If g is cyclic then so is $h^{-1} \circ g$. Hence if ω is cyclic then so is the new weighting. We repeat this construction for every non-idempotent operation in $\text{supp}(\omega)$. Finally, we obtain an idempotent weighted polymorphism ω' which satisfies the conditions of the lemma.

Proposition 22. *Let Γ be a valued constraint language which is a core. Then $\text{IdPol}^+(\Gamma) = \text{Pol}^+(\Gamma_c)$.*

Proof. Clearly both sets contain all the projections. Let us take $f \in \text{Pol}^+(\Gamma_c)$ that is not a projection and let ω be a weighted polymorphism of Γ_c such that $f \in \text{supp}(\omega)$. There is a corresponding weighted polymorphism ω' of Γ , which is equal to ω on the idempotent operations and equal 0 otherwise. Then we have $f \in \text{supp}(\omega')$. Since f is idempotent it follows that $f \in \text{IdPol}^+(\Gamma)$.

To prove the reverse inclusion consider $f \in \text{IdPol}^+(\Gamma)$ that is not a projection. Let ω be a weighted polymorphism of Γ such that $f \in \text{supp}(\omega)$. By Lemma 21 there exists an idempotent weighted polymorphism ω' of Γ such that $\text{supp}(\omega) \cap \text{IdPol}(\Gamma) \subseteq \text{supp}(\omega')$. The weighting ω'' , defined as a restriction of ω' to the idempotent operations, is a weighted polymorphism of Γ_c with $f \in \text{supp}(\omega'')$. Hence $f \in \text{Pol}^+(\Gamma_c)$.

4 Weighted varieties

One of the fundamental results of the algebraic approach to CSP [3, 7, 20] says that the complexity of a crisp language Γ depends only on the variety generated by the algebra $(D, \text{Pol}(\Gamma))$. We generalize this fact to VCSP.

A k -ary *weighting* ω of an algebra \mathbf{A} is a function that assigns rational weights to all k -ary term operations of \mathbf{A} in such a way, that the sum of all

weights is 0, and if $\omega(f) < 0$ then f is a projection. A (*proper*) *superposition* $\omega[g_1, \dots, g_k]$ of a weighting ω with a list of l -ary term operations g_1, \dots, g_k from \mathbf{A} is defined the same way as for clones (see Definition 6). An algebra \mathbf{A} together with a set of weightings closed under non-negative scaling, addition of weightings of equal arity and proper superposition with operations from \mathbf{A} is called a *weighted algebra*.

For a variety \mathcal{V} over a signature Σ and a term t we denote by $[t]_{\mathcal{V}}$ the equivalence class of t under the relation $\approx_{\mathcal{V}}$ such that $t \approx_{\mathcal{V}} s$ if and only if the variety \mathcal{V} satisfies the identity $t \approx s$ (we skip the subscript, writing $[t]$ instead $[t]_{\mathcal{V}}$, whenever the variety is clear from the context). Observe that if the variety is locally finite then there are finitely many equivalence classes of terms of a fixed arity [8].

Definition 23. *Let \mathcal{V} be a locally finite variety over a signature Σ . A k -ary weighting ω of \mathcal{V} is a function that assigns rational weights to all equivalence classes of k -ary terms over Σ in such a way, that the sum of all weights is 0, and if $\omega([t]) < 0$ then \mathcal{V} satisfies the identity $t(x_1, \dots, x_k) \approx x_i$ for some $i \in \{1, \dots, k\}$. The variety \mathcal{V} together with a nonempty set of weightings is called a weighted variety.*

Take any finite algebra $\mathbf{B} \in \mathcal{V}$. A k -ary weighting ω of \mathcal{V} induces a weighting $\omega^{\mathbf{B}}$ of \mathbf{B} in a natural way:

$$\omega^{\mathbf{B}}(f) = \sum_{\{[t] \mid t^{\mathbf{B}}=f\}} \omega([t]).$$

If $\omega([t]) < 0$ then the term operation $t^{\mathbf{B}}$ is a projection, and hence the weighting $\omega^{\mathbf{B}}$ is proper. For a weighted variety \mathcal{V} , by $\mathbf{B} \in \mathcal{V}$ we mean the algebra \mathbf{B} together with the set of weightings induced by \mathcal{V} .

For every weighting ω of a finite weighted algebra \mathbf{A} there is a corresponding weighting ω of the variety $\mathcal{V}(\mathbf{A})$ defined by $\omega([t]) = \omega(t^{\mathbf{A}})$. It follows from Birkhoff's theorem (see Theorem 11) that it is well defined. A weighted variety $\mathcal{V}(\mathbf{A})$ generated by a weighted algebra \mathbf{A} is the variety $\mathcal{V}(\mathbf{A})$ together with the set of weightings corresponding to the weightings of \mathbf{A} . The correspondence is one-to-one so for simplicity we often identify the weightings of $\mathcal{V}(\mathbf{A})$ with the weightings of \mathbf{A} .

We prove that every finite algebra $\mathbf{B} \in \mathcal{V}(\mathbf{A})$ together with the set of weightings induced by $\mathcal{V}(\mathbf{A})$ is a weighted algebra. It is straightforward to check its closure under non-negative scaling and addition of weightings of equal arity. We only need to show that \mathbf{B} is closed under proper superpositions.

Proposition 24. *For a finite weighted algebra \mathbf{A} over a fixed signature Σ and a finite algebra $\mathbf{B} \in \mathcal{V}(\mathbf{A})$ let $\omega^{\mathbf{B}}$ be a k -ary weighting of \mathbf{B} induced by the weighted variety $\mathcal{V}(\mathbf{A})$. If for some list $f_1^{\mathbf{B}}, \dots, f_k^{\mathbf{B}}$ of l -ary term operations from \mathbf{B} the weighting $\omega^{\mathbf{B}}[f_1^{\mathbf{B}}, \dots, f_k^{\mathbf{B}}]$ is proper then it is induced by some weighting of $\mathcal{V}(\mathbf{A})$.*

In the proof we use Gordan's Theorem (which is a straightforward consequence of Lemma 17).

Theorem 25 (Gordan [13]). *Let S and T be finite sets of indices. For all $i \in S$, and all $j \in T$, let $a_{i,j}$ be rational numbers. Exactly one of the following holds:*

- *Either there exists a set of non-negative rational numbers $\{z_i \mid i \in S\}$ such that*

$$\begin{aligned} & \text{for some } i \in S, \quad z_i > 0, \\ & \text{for each } j \in T, \quad \sum_{i \in S} a_{i,j} z_i = 0. \end{aligned}$$

- *Or else there exists a set of rational numbers $\{y_j \mid j \in T\}$ such that*

$$\text{for each } i \in S, \quad \sum_{j \in T} y_j a_{i,j} > 0.$$

Proof. (of Proposition 24) Let $\omega^{\mathbf{B}}$ and $f_1^{\mathbf{B}}, \dots, f_k^{\mathbf{B}}$ be as in the statement of the proposition. Assume that the weighting $\omega^{\mathbf{B}}[f_1^{\mathbf{B}}, \dots, f_k^{\mathbf{B}}]$ is proper.

Notice that the following conditions are equivalent:

- the operation $f_i^{\mathbf{B}}$ is the projection π_j on the j -th coordinate,
- there exists a term t such that $t^{\mathbf{B}} = f_i^{\mathbf{B}}$ and $t^{\mathbf{A}}$ is the projection π_j on the j -th coordinate.

For each $i \in \{1, \dots, k\}$ consider the set F_i of equivalence classes of terms over Σ defined by

$$F_i = \begin{cases} \{[t] \mid t^{\mathbf{A}} = \pi_j\} & \text{if } f_i^{\mathbf{B}} = \pi_j, \\ \{[t] \mid t^{\mathbf{B}} = f_i^{\mathbf{B}}\} & \text{otherwise} \end{cases}$$

(observe that if $f_i^{\mathbf{B}}$ is a projection then F_i contains a single equivalence class). Take ω to be some k -ary weighting of \mathbf{A} that induces $\omega^{\mathbf{B}}$, and let

$$W = \{\omega[t_1^{\mathbf{A}}, \dots, t_k^{\mathbf{A}}] \mid [t_i] \in F_i\}.$$

Suppose that for some choice of equivalence classes $[t_i] \in F_i$ the superposition $\omega[t_1^{\mathbf{A}}, \dots, t_k^{\mathbf{A}}]$ is proper. The weighting $\omega[t_1^{\mathbf{A}}, \dots, t_k^{\mathbf{A}}]$ of \mathbf{A} induces a weighting of \mathbf{B} which is equal to $\omega^{\mathbf{B}}[f_1^{\mathbf{B}}, \dots, f_k^{\mathbf{B}}]$, thus in this case the proof is concluded.

This shows that a superposition of $\omega^{\mathbf{B}}$ with any list of projections is always induced by some weighting of \mathbf{A} .

Now let us deal with the case when none of the weightings in W is proper. Without loss of generality we can assume that the operations $f_1^{\mathbf{B}}, \dots, f_k^{\mathbf{B}}$ are pairwise distinct (otherwise we replace $\omega^{\mathbf{B}}$ by its superposition with a suitable list of projections) and hence the sets F_i are disjoint. Let $F = \bigcup F_i$. We remove from F the element of F_i if $f_i^{\mathbf{B}}$ is a projection. The removed elements cannot cause a problem and therefore we assume that for every $[t] \in F$ the operation $t^{\mathbf{A}}$ is not a projection. We apply Gordan's Theorem to the following system of linear equations:

$$\sum_{\nu \in W} \nu(t^{\mathbf{A}}) \cdot z_\nu - z_{[t]} = 0, \text{ for each } [t] \in F.$$

If this system has a non-zero solution in non-negative rational numbers then $z_\nu > 0$ for some $\nu \in W$. Observe that the weighting $v = \sum_{\nu \in W} \nu \cdot z_\nu$ is proper. Indeed, by the definition of a superposition the only non-projections that could be assigned negative weights by v are the operations $t^{\mathbf{A}}$ where $[t] \in F$. But each such operation $t^{\mathbf{A}}$ is assigned a non-negative weight $z_{[t]}$. Hence, by Lemma 13 the weighting v is equal to a proper superposition of some weighting of \mathbf{A} with a list of l -ary term operations of \mathbf{A} . Finally, let $p = \sum_{\nu \in W} z_\nu > 0$. The weighting $\frac{1}{p}v$ of \mathbf{A} induces a weighting of \mathbf{B} which is equal to $\omega^{\mathbf{B}}[f_1^{\mathbf{B}}, \dots, f_k^{\mathbf{B}}]$.

Otherwise, there exists a set $\{y_{[t]} \mid [t] \in F\}$ of rational numbers, such that

$$\text{for each } \nu \in W, \quad \sum_{[t] \in F} y_{[t]} \cdot \nu(t^{\mathbf{A}}) > 0,$$

and $y_{[t]} < 0$ for each $[t] \in F$. For every $i \in \{1, \dots, k\}$ let us choose $[t_i] \in F_i$ satisfying $y_{[t_i]} = \max\{y_{[t]} \mid [t] \in F_i\}$ (if $f_i^{\mathbf{B}}$ is a projection then we choose $[t_i] \in F_i$ to be the only element of F_i and put $y_{[t_i]} = 0$) and consider the weighting $v = \omega[t_1^{\mathbf{A}}, \dots, t_k^{\mathbf{A}}]$. Notice that v may assign negative weights only to operations $t_i^{\mathbf{A}}$. Since

$$\sum_{[t] \in F_1} y_{[t]} \cdot v(t^{\mathbf{A}}) + \dots + \sum_{[t] \in F_k} y_{[t]} \cdot v(t^{\mathbf{A}}) > 0,$$

then $\sum_{[t] \in F_i} y_{[t]} \cdot v(t^{\mathbf{A}}) > 0$ for some $i \in \{1, \dots, k\}$. Hence

$$0 < \sum_{[t] \in F_i} y_{[t]} \cdot v(t^{\mathbf{A}}) \leq \sum_{[t] \in F_i} y_{[t_i]} \cdot v(t^{\mathbf{A}}) = y_{[t_i]} \cdot \sum_{[t] \in F_i} v(t^{\mathbf{A}}).$$

It follows that $\sum_{[t] \in F_i} v(t^{\mathbf{A}}) < 0$, which is a contradiction, since $\sum_{[t] \in F_i} v(t^{\mathbf{A}})$ is the weight that the proper weighting $\omega^{\mathbf{B}}[f_1^{\mathbf{B}}, \dots, f_k^{\mathbf{B}}]$ assigns to the operation $f_i^{\mathbf{B}}$ (which is not a projection).

For a finite weighted algebra \mathbf{A} let $\text{Imp}(\mathbf{A})$ denote the set of those cost functions on A that are improved by all weightings of \mathbf{A} . We prove that for each finite weighted algebra $\mathbf{B} \in \mathcal{V}(\mathbf{A})$ the valued constraint language $\text{Imp}(\mathbf{B})$ is not harder than $\text{Imp}(\mathbf{A})$. The proof consists of a sequence of lemmas.

Lemma 26. *Let \mathbf{A} be a finite weighted algebra. For any $\mathbf{B} \in P_{fin}(\mathbf{A})$, there is a polynomial-time reduction of $\text{VCSP}(\text{Imp}(\mathbf{B}))$ to $\text{VCSP}(\text{Imp}(\mathbf{A}))$.*

Proof. Let A^n be the universe of \mathbf{B} and let Γ be a finite subset of $\text{Imp}(\mathbf{B})$. Take $\varrho \in \Gamma$ to be an r -ary cost function. There is a natural way of defining a corresponding cost function of arity $n \cdot r$ on the set A . We denote this cost function by ϱ' .

Let ω be a k -ary weighting of the weighted algebra \mathbf{A} . The corresponding k -ary weighting $\omega^{\mathbf{B}}$ of \mathbf{B} is a weighted polymorphism of ϱ . Then it is not hard to show that ω is a weighted polymorphism of ϱ' , as the basic operations of \mathbf{B}

are the operations of \mathbf{A} computed coordinate-wise. Hence, each weighting of \mathbf{A} is a weighted polymorphism of ϱ' , which means that $\varrho' \in \text{Imp}(\mathbf{A})$.

For each $\varrho \in \Gamma$ we have defined a corresponding $\varrho' \in \text{Imp}(\mathbf{A})$. Let $\Gamma' \subseteq \text{Imp}(\mathbf{A})$ be the (finite) set of all those cost functions.

Now take an arbitrary instance $\mathcal{I} = (V, A^n, \mathcal{C})$ of $\text{VCSP}(\Gamma)$. Replace the domain A^n by A , and each variable $v_i \in V$ by a set of n variables $\{v_i^1, \dots, v_i^n\}$, obtaining a new set of variables V' . In each constraint $(\sigma, \varrho) \in \mathcal{C}$, where ϱ is an r -ary cost function, replace the r -tuple σ of variables from V by the corresponding nr -tuple of variables from V' , and the cost function ϱ by the corresponding cost function ϱ' from Γ' . The new instance $\mathcal{I}' = (V', A, \mathcal{C}')$ is an instance of $\text{VCSP}(\Gamma')$. It is easy to see that there is a one-to-one correspondence between the optimal assignments for \mathcal{I} and the optimal assignments for \mathcal{I}' .

Lemma 27. *Let \mathbf{A} be a finite weighted algebra. For any $\mathbf{B} \in \mathcal{S}(\mathbf{A})$, there is a polynomial-time reduction of $\text{VCSP}(\text{Imp}(\mathbf{B}))$ to $\text{VCSP}(\text{Imp}(\mathbf{A}))$.*

Notice that $\text{Imp}(\mathbf{B}) \subseteq \text{Imp}(\mathbf{A})$, so there is nothing to be proved.

Lemma 28. *Let \mathbf{A} be a finite weighted algebra. For any $\mathbf{B} \in \mathcal{H}(\mathbf{A})$, there is a polynomial-time reduction of $\text{VCSP}(\text{Imp}(\mathbf{B}))$ to $\text{VCSP}(\text{Imp}(\mathbf{A}))$.*

Proof. By the isomorphism theorem we can consider \mathbf{B} to be a quotient algebra \mathbf{A}/\sim rather than a homomorphic image of \mathbf{A} . Let A/\sim be the universe of \mathbf{B} and let Γ be a finite subset of $\text{Imp}(\mathbf{B})$. Take $\varrho \in \Gamma$ to be a r -ary cost function. We define a corresponding cost function ϱ' of arity r on the set A by $\varrho'(x_1, \dots, x_r) = \varrho([x_1]_{\sim}, \dots, [x_r]_{\sim})$.

Let ω be a k -ary weighting of the weighted algebra \mathbf{A} . The corresponding k -ary weighting $\omega^{\mathbf{B}}$ of \mathbf{B} is a weighted polymorphism of ϱ . It is not hard to show that ω is a weighted polymorphism of ϱ' . Hence, each weighting of \mathbf{A} is a weighted polymorphism of ϱ' , which means that $\varrho' \in \text{Imp}(\mathbf{A})$.

For each $\varrho \in \Gamma$ we have defined a corresponding $\varrho' \in \text{Imp}(\mathbf{A})$. Let $\Gamma' \subseteq \text{Imp}(\mathbf{A})$ be the (finite) set of all those cost functions.

Now take an arbitrary instance $\mathcal{I} = (V, A/\sim, \mathcal{C})$ of $\text{VCSP}(\Gamma)$. Replace the domain A/\sim by A . In each constraint $(\sigma, \varrho) \in \mathcal{C}$ replace the cost function ϱ by a corresponding cost function ϱ' from Γ' . The new instance $\mathcal{I}' = (V, A, \mathcal{C}')$ is an instance of $\text{VCSP}(\Gamma')$.

If $s': V \rightarrow A$ is an optimal assignment for \mathcal{I}' , then $s: V \rightarrow A/\sim$ defined by $s(v) = [s'(v)]_{\sim}$ is an optimal assignment for \mathcal{I} . On the other hand, if $s: V \rightarrow A/\sim$ is an optimal assignment for \mathcal{I} , then any assignment $s': V \rightarrow A$, such that for each $v \in V$, we have $s'(v) \in s(v)$, is optimal for \mathcal{I}' .

The above lemmas together with Proposition 10, imply the following:

Proposition 29. *For any finite weighted algebra \mathbf{A} , and any finite $\mathbf{B} \in \mathcal{V}(\mathbf{A})$, there is a polynomial-time reduction of $\text{VCSP}(\text{Imp}(\mathbf{B}))$ to $\text{VCSP}(\text{Imp}(\mathbf{A}))$.*

For a valued constraint language Γ the weighted algebra $(D, \text{wPol}(\Gamma))$ is the algebra $(D, \text{Pol}(\Gamma))$ together with the set of weightings $\text{wPol}(\Gamma)$. By Proposition 29 the complexity of Γ depends only on the weighted variety generated by the weighted algebra $(D, \text{wPol}(\Gamma))$.

5 Dichotomy conjecture

An operation t of arity k is called a *Taylor operation* of an algebra (or a variety), if t is idempotent and for every $j \leq k$ it satisfies an identity of the form

$$t(\square_1, \square_2, \dots, \square_k) \approx t(\triangle_1, \triangle_2, \dots, \triangle_k),$$

where all \square_i s and \triangle_i s are substituted with either x or y , but \square_j is x whenever \triangle_j is y . In this section we work towards a proof of the following theorem:

Theorem 30. *Let Γ be a finite core valued constraint language. If $\text{Pol}^+(\Gamma)$ does not have a Taylor operation, then Γ is NP-hard.*

We conjecture³ that these are the only cases of finite core languages which give rise to NP-hard VCSPs.

Conjecture. *Let Γ be a finite core valued constraint language. If $\text{Pol}^+(\Gamma)$ does not have a Taylor operation, then Γ is NP-hard. Otherwise it is tractable.*

For crisp languages $\text{Pol}^+(\Gamma) = \text{Pol}(\Gamma)$. Therefore Theorem 30 generalizes the well-known result of Bulatov, Jeavons and Krokhin [3, 7] concerning crisp core languages. Similarly the above conjecture is a generalization of The Algebraic Dichotomy Conjecture for CSP. Later on we show that it is supported by all known partial results on the complexity of VCSPs.

To prove Theorem 30 we use the following characterization of algebras possessing a Taylor operation:

Theorem 31 (Taylor [24]). *Let \mathbf{A} be a finite idempotent algebra, then the following are equivalent:*

- \mathbf{A} has a Taylor operation,
- $\mathcal{V}(\mathbf{A})$ (equivalently $\text{HS}(\mathbf{A})$) does not contain a two-element algebra whose every term operation is a projection.

First let us prove an auxiliary lemma.

Lemma 32. *Let Γ be a finite core valued constraint language over a domain D , and let R be an r -ary relation which is compatible with every polymorphism from $\text{Pol}^+(\Gamma)$. Then there exists a cost function ϱ_R in $\text{wRelClo}(\Gamma)$, and a rational number P , such that for every r -tuple \mathbf{x} the following conditions are satisfied:*

- $\varrho_R(\mathbf{x}) \geq P$ and
- $\varrho_R(\mathbf{x}) = P$ if and only if $\mathbf{x} \in R$.

Proof. Let $R = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ be a relation as in the statement of the lemma. By Proposition 18 there exists a cost function $\varrho': \mathcal{O}_D^{(m)} \rightarrow \overline{\mathbb{Q}}$ in $\text{wRelClo}(\Gamma)$, and a rational number P , such that for every $f \in \mathcal{O}_D^{(m)}$:

³ The conjecture was suggested in a conversation by Libor Barto, however it might have appeared independently earlier.

- $\varrho'(f) \geq P$,
- $\varrho'(f) < \infty$ if and only if $f \in \text{Pol}(\Gamma)$,
- $\varrho'(f) = P$ if and only if $f \in \text{Pol}^+(\Gamma)$.

Consider the coordinates $\mathbf{b}_1, \dots, \mathbf{b}_r$, such that $\mathbf{b}_i(j) = \mathbf{x}_j(i)$. Minimising the cost function ϱ' over all the other coordinates we obtain a cost function ϱ satisfying the given conditions.

Proof. (of Theorem 30) Let Γ be a finite core valued constraint language over a domain D , and let Γ_c be a rigid core of Γ as defined in Subsection 3.3. Suppose that $\text{Pol}^+(\Gamma)$ does not have a Taylor operation. By Proposition 22 we have that $\text{IdPol}^+(\Gamma) = \text{Pol}^+(\Gamma_c)$. Therefore, $\text{Pol}^+(\Gamma_c)$ does not have a Taylor operation. Below we prove that $\text{VCSP}(\Gamma_c)$ is NP-hard. This implies, by Proposition 20, that $\text{VCSP}(\Gamma)$ is NP-hard which concludes the proof.

Let \mathbf{A} denote the idempotent algebra $\text{Pol}^+(\Gamma_c)$ over the universe D . By Taylor's theorem $\text{HS}(\mathbf{A})$ contains a two-element algebra \mathbf{B} whose every term operation is a projection. By the isomorphism theorem we can consider \mathbf{B} to be a quotient algebra rather than a homomorphic image of a subalgebra of \mathbf{A} . In other words, there exists a binary relation S compatible with \mathbf{A} , which is an equivalence relation on some subuniverse D' of D and has two equivalence classes $[d_0]_S$ and $[d_1]_S$. Moreover, the term operations defined on the set of equivalence classes of S using their arbitrarily chosen representatives are all projections.

Every relation is compatible with a two-element algebra whose every term operation is a projection. Consider the relation $R = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$. It corresponds to the ONE-IN-THREE SAT problem, which is NP-complete [21]. We define a ternary relation S_{IN3} on D' by:

$$S_{\text{IN3}} = \{(x_1, x_2, x_3) : \text{exactly one of } x_1, x_2, x_3 \text{ belongs to } [d_1]_S\}.$$

This relation is compatible with \mathbf{A} . Hence there exists a cost function $\varrho_{S_{\text{IN3}}}$ in $\text{wRelClo}(\Gamma_c)$ satisfying the conditions given by Lemma 32.

Now, for every instance of ONE-IN-THREE SAT it is easy to construct (in polynomial time) an instance of $\text{VCSP}(\Gamma_c)$ such that if the instance of ONE-IN-THREE SAT has a solution then this solution gives rise to the minimal evaluation of the constructed instance. This finishes the reduction.

As the Taylor operation is difficult to work with, in the following we use a characterization of Taylor algebras as the algebras possessing a cyclic term.

Theorem 33 (Barto and Kozik [1]). *A finite idempotent algebra \mathbf{A} has a Taylor operation if and only if it has a cyclic operation if and only if it has a cyclic operation of arity p , for every prime $p > |\mathbf{A}|$.*

Since $(D, \text{IdPol}^+(\Gamma))$ is a finite idempotent algebra it follows that:

Corollary 34. *For any finite core valued constraint language Γ , $\text{Pol}^+(\Gamma)$ has a Taylor operation if and only if it has an idempotent cyclic operation.*

5.1 Two-element domain

A complete complexity classification for valued constraint languages over a two-element domain was established in [10]. All tractable languages have been defined via multimorphisms, which are a more restricted form of weighted polymorphisms. A k -ary *multimorphism* of a language Γ , specified as a k -tuple $\langle f_1, \dots, f_k \rangle$ of k -ary operations on D , is a k -ary weighted polymorphism ω of Γ such that $\omega = \frac{1}{k} (\sum_i f_i - \sum_i \pi_i^{(k)})$.

An operation $f \in \mathcal{O}_D^{(3)}$ is called a *majority* operation if for every $x, y \in D$ we have that $f(x, x, y) = f(x, y, x) = f(y, x, x) = x$. Similarly, an operation $f \in \mathcal{O}_D^{(3)}$ is called a *minority* operation if for every $x, y \in D$ it satisfies $f(x, x, y) = f(x, y, x) = f(y, x, x) = y$. Observe that on a two-element domain there is precisely one majority operation, which we denote by Mjrty, and precisely one minority operation, which we denote by Mnrty.

Theorem 35 (Cohen et al. [10]). *Let Γ be a core valued constraint language on $D = \{0, 1\}$. If Γ admits at least one of the following six multimorphisms, then Γ is tractable. Otherwise it is NP-hard.*

1. $\langle \min, \min \rangle$,
2. $\langle \max, \max \rangle$,
3. $\langle \min, \max \rangle$,
4. $\langle \text{Mjrty}, \text{Mjrty}, \text{Mjrty} \rangle$,
5. $\langle \text{Mnrty}, \text{Mnrty}, \text{Mnrty} \rangle$,
6. $\langle \text{Mjrty}, \text{Mjrty}, \text{Mnrty} \rangle$.

In [10] the complexity classification is given for languages that are not necessarily cores. It is not difficult to prove, though, that the general case is equivalent to the above theorem. This is because every language over a two-element domain which is not a core is tractable.

We show that the dichotomy conjecture for VCSP agrees with the complexity classification for valued constraint languages over a two-element domain.

Proposition 36. *Let Γ be a finite core valued constraint language on $D = \{0, 1\}$. Then $\text{Pol}^+(\Gamma)$ has an idempotent cyclic operation if and only if Γ admits at least one of the following six multimorphisms.*

1. $\langle \min, \min \rangle$,
2. $\langle \max, \max \rangle$,
3. $\langle \min, \max \rangle$,
4. $\langle \text{Mjrty}, \text{Mjrty}, \text{Mjrty} \rangle$,
5. $\langle \text{Mnrty}, \text{Mnrty}, \text{Mnrty} \rangle$,
6. $\langle \text{Mjrty}, \text{Mjrty}, \text{Mnrty} \rangle$.

On $D = \{0, 1\}$ there are precisely two constant operations, which we denote by Const_0 and Const_1 . By Inv we denote the *inversion* operation defined by $\text{Inv}(0) = 1$ and $\text{Inv}(1) = 0$. To prove Proposition 36 we use the following theorem:

Theorem 37 (Cohen et al. [9]). *Let W be a weighted clone on $D = \{0, 1\}$ that contains a weighting which assigns positive weight to at least one operation that is not a projection. Then W contains one of the following nine weightings:*

1. $\langle \text{Const}_0 \rangle$,
2. $\langle \text{Const}_1 \rangle$,
3. $\langle \text{Inv} \rangle$,
4. $\langle \text{min}, \text{min} \rangle$,
5. $\langle \text{max}, \text{max} \rangle$,
6. $\langle \text{min}, \text{max} \rangle$,
7. $\langle \text{Mjrty}, \text{Mjrty}, \text{Mjrty} \rangle$,
8. $\langle \text{Mnrty}, \text{Mnrty}, \text{Mnrty} \rangle$,
9. $\langle \text{Mjrty}, \text{Mjrty}, \text{Mnrty} \rangle$.

Proof. (of Proposition 36) Each of the operations min , max , Mjrty and Mnrty is idempotent and cyclic. If Γ admits at least one of the six multimorphisms listed in the statement of the proposition then obviously $\text{Pol}^+(\Gamma)$ has an idempotent cyclic operation.

For the other direction, let Γ be a finite core valued constraint language on $D = \{0, 1\}$ such that $\text{Pol}^+(\Gamma)$ has an idempotent cyclic operation. Let Γ_c be a rigid core of Γ as defined in Subsection 3.3. By Proposition 22 we have that $\text{IdPol}^+(\Gamma) = \text{Pol}^+(\Gamma_c)$. Therefore, the weighted clone $\text{wPol}(\Gamma_c)$ contains a weighting which assigns positive weight to at least one operation that is not a projection. Then $\text{wPol}(\Gamma_c)$ contains one of the nine weightings listed in Theorem 37. Since the first three of them are not idempotent, it follows that Γ_c , and hence Γ , admits one of the six remaining multimorphisms, which finishes the proof.

5.2 Finite-valued languages

Theorem 38 (Thapper and Živný [25]). *Let Γ be a finite-valued constraint language which is a core. If Γ admits an idempotent cyclic weighted polymorphism of some arity $m > 1$, then Γ is tractable. Otherwise it is NP-hard.*

To show that our conjecture agrees with the above complexity classification we prove the following result (which holds for general-valued languages):

Proposition 39. *Let Γ be a core valued constraint language. Then Γ admits an idempotent cyclic weighted polymorphism of some arity $m > 1$ if and only if $\text{Pol}^+(\Gamma)$ contains an idempotent cyclic operation of the same arity.*

One implication is straightforward: if Γ admits an idempotent cyclic weighted polymorphism of some arity $m > 1$ then $\text{Pol}^+(\Gamma)$ contains an idempotent cyclic operation. To show the other implication we use a technique of constructing weighted polymorphism introduced in [18].

The construction of a new weighted polymorphism of arity m is based on grouping operations in $\mathcal{O}_D^{(m)}$ into so-called *collections* and working with weightings that assign the same weight to every operation in a collection.

Let \mathbb{G} be a fixed set of collections, i.e., subsets of $\mathcal{O}_D^{(m)}$, and let $\mathbb{G}^* \subseteq \mathbb{G}$ be a set of collections satisfying some desired property. An *expansion operator* Exp takes a collection $\mathbf{g} \in \mathbb{G}$ and produces a probability distribution δ over \mathbb{G} . We say that Exp is *valid* for a language Γ if, for any $\varrho \in \Gamma$ and any $\mathbf{g} \in \mathbb{G}$, the probability distribution $\delta = \text{Exp}(\mathbf{g})$ satisfies

$$\sum_{\mathbf{h} \in \mathbb{G}} \sum_{h \in \mathbf{h}} \frac{\delta(\mathbf{h})}{|\mathbf{h}|} \varrho(h(\mathbf{x}_1, \dots, \mathbf{x}_m)) \leq \sum_{g \in \mathbf{g}} \frac{1}{|\mathbf{g}|} \varrho(g(\mathbf{x}_1, \dots, \mathbf{x}_m)),$$

for any $\mathbf{x}_1, \dots, \mathbf{x}_m \in \text{Feas}(\varrho)$. We say that the operator Exp is *non-vanishing* (with respect to the pair $(\mathbb{G}, \mathbb{G}^*)$) if, for any $\mathbf{g} \in \mathbb{G}$, there exists a sequence of collections $\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_r$ with $\mathbf{g}_0 = \mathbf{g}$, such that for each $i \in \{0, \dots, r-1\}$ the collection \mathbf{g}_{i+1} is assigned a non-zero probability by $\text{Exp}(\mathbf{g}_i)$, and $\mathbf{g}_r \in \mathbb{G}^*$.

Lemma 40 (Expansion Lemma [18]). *Let Exp be an expansion operator which is valid for the language Γ and non-vanishing with respect to $(\mathbb{G}, \mathbb{G}^*)$. If Γ admits a weighted polymorphism ω with $\text{supp}(\omega) \subseteq \bigcup \mathbb{G}$, then it also admits a weighted polymorphism ω^* with $\text{supp}(\omega^*) \subseteq \bigcup \mathbb{G}^*$.*

Proof. (of Proposition 39) In order to show the remaining implication assume that $\text{Pol}^+(\Gamma)$ contains an idempotent cyclic operation f of arity $m > 1$. There exists a weighted polymorphism ω of Γ such that $f \in \text{supp}(\omega)$.

We define \sim to be the smallest equivalence relation on $\mathcal{O}_D^{(m)}$ such that $g \sim g'$ if $g(x_1, x_2, \dots, x_k) = g'(x_2, \dots, x_k, x_1)$. Observe that if $g \sim g'$ and $g \in \text{Pol}(\Gamma)$ then also $g' \in \text{Pol}(\Gamma)$. Let \mathbb{G} consist of the equivalence classes of the relation \sim restricted to $\text{Pol}(\Gamma)$, and let $\mathbb{G}^* \subseteq \mathbb{G}$ be the set of all one-element equivalence classes, i.e., each $\mathbf{g} \in \mathbb{G}^*$ contains a single cyclic operation.

We now define the expansion operator Exp . Take an arbitrary $\mathbf{g} \in \mathbb{G} \setminus \mathbb{G}^*$ (for $\mathbf{g} \in \mathbb{G}^*$ we produce the probability distribution choosing \mathbf{g} with probability 1) and choose a single operation $g \in \mathbf{g}$. Notice that $\mathbf{g} = \{g_1, g_2, \dots, g_m\}$, where $g_1 = g$ and $g_i(x_1, \dots, x_n) = g(x_i, x_{i+1}, \dots, x_{i-1})$ for $i \in \{2, \dots, m\}$. Consider a weighting

$$\nu = c \cdot (\omega[g_1, g_2, \dots, g_m] + \omega[g_2, \dots, g_m, g_1] + \dots + \omega[g_m, g_1, \dots, g_{m-1}]),$$

where c is a suitable positive rational, which we define later on⁴.

The weighting ν assigns a positive weight to a cyclic operation $f[g_1, g_2, \dots, g_m]$. This proves that ν is not zero-valued, and hence in the above definition c can be chosen so that the sum of positive weights in ν equals 1. We say that a weighting ω is *weight-symmetric* if $\omega(g) = \omega(g')$ whenever $g \sim g'$. It is easy to check that ν is weight-symmetric. We define $\text{Exp}(\mathbf{g})$ to be a probability distribution δ on \mathbb{G} such that

$$\delta(\mathbf{h}) = \begin{cases} |\mathbf{h}| \cdot \nu(h) & \text{if } \mathbf{h} \subseteq \text{supp}(\nu), \\ 0 & \text{otherwise,} \end{cases}$$

⁴ Note that the definition of ν does not depend on the choice of g from \mathbf{g} .

where h is any of the operations in \mathbf{h} . We have already pointed out that ν assigns a positive weight to a cyclic operation $f[g_1, g_2, \dots, g_m]$. It follows that $\text{Exp}(\mathbf{g})$ assigns non-zero probability to the one-element equivalence class $\{f[g_1, g_2, \dots, g_m]\}$. Therefore, Exp is non-vanishing.

It remains to show that Exp is valid for Γ . Observe that ν assigns negative weights only to the operations in \mathbf{g} . Since it is weight-symmetric, $\nu(g) = -\frac{1}{|\mathbf{g}|}$ for every $g \in \mathbf{g}$. The weighting ν might not be valid but it is not difficult to see that it satisfies the condition characterizing weighted polymorphisms, i.e., for any cost function $\varrho \in \Gamma$, and any list of tuples $\mathbf{x}_1, \dots, \mathbf{x}_m \in \text{Feas}(\varrho)$, we have

$$\begin{aligned} \sum_{f \in \text{Pol}_m(\Gamma)} \nu(f) \cdot \varrho(f(\mathbf{x}_1, \dots, \mathbf{x}_m)) &\leq 0, \text{ hence} \\ \sum_{h \in \text{supp}(\nu)} \nu(h) \cdot \varrho(h(\mathbf{x}_1, \dots, \mathbf{x}_m)) &\leq \sum_{g \in \mathbf{g}} \frac{1}{|\mathbf{g}|} \cdot \varrho(g(\mathbf{x}_1, \dots, \mathbf{x}_m)), \text{ but} \\ \sum_{h \in \text{supp}(\nu)} \nu(h) \cdot \varrho(h(\mathbf{x}_1, \dots, \mathbf{x}_m)) &= \sum_{\mathbf{h} \in \mathbf{G}} \sum_{h \in \mathbf{h}} \frac{\delta(\mathbf{h})}{|\mathbf{h}|} \cdot \varrho(h(\mathbf{x}_1, \dots, \mathbf{x}_m)). \end{aligned}$$

This proves that Exp is valid, so by Lemma 40 the language Γ admits a weighted polymorphism ω^* whose support contains only cyclic m -ary operations. Moreover, it follows from the proof of the Expansion Lemma in [18] that ω^* can be constructed so that $f \in \text{supp}(\omega^*)$. By Lemma 21 there exists an idempotent weighted polymorphism ω' of Γ such that $\text{supp}(\omega^*) \cap \text{IdPol}(\Gamma) \subseteq \text{supp}(\omega')$. Its support is non-empty and contains cyclic operations only. This concludes the proof.

5.3 Conservative languages

A valued constraint language Γ over a domain D is called *conservative* if it contains all $\{0, 1\}$ -valued unary cost functions on D . An operation $f \in \mathcal{O}_D^{(k)}$ is *conservative* if for every $x_1, \dots, x_k \in D$ we have that $f(x_1, \dots, x_k) \in \{x_1, \dots, x_k\}$, and a weighted polymorphism is *conservative* if its support contains conservative operations only.

A *Symmetric Tournament Pair (STP)* is a conservative binary multimorphism $\langle \sqcap, \sqcup \rangle$, where both operations are commutative, i.e., $\sqcap(x, y) = \sqcap(y, x)$ and $\sqcup(x, y) = \sqcup(y, x)$ for all $x, y \in D$, and moreover $\sqcap(x, y) \neq \sqcup(x, y)$ for all $x \neq y$. A *MJN* is a ternary conservative multimorphism $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$, such that Mj_1, Mj_2 are majority operations, and Mn_3 is a minority operation.

Theorem 41 (Kolmogorov and Živný [19]). *Let Γ be a conservative constraint language over a domain D . If Γ admits a conservative binary multimorphism $\langle \sqcap, \sqcup \rangle$ and a conservative ternary multimorphism $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$, and there is a family M of two-element subsets of D , such that:*

- for every $\{x, y\} \in M$, $\langle \sqcap, \sqcup \rangle$ restricted to $\{x, y\}$ is an STP,
- for every $\{x, y\} \notin M$, $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$ restricted to $\{x, y\}$ is an MJN,

then Γ is tractable. Otherwise it is NP-hard.

Observe that every weighted polymorphism of a conservative language Γ is conservative. Indeed, consider a k -ary weighted polymorphism $\omega \in \text{wPol}(\Gamma)$ and take any $x_1, \dots, x_k \in D$. Let $\varrho \in \Gamma$ be a unary cost function such that $\varrho(x_i) = 0$ for $i \in \{1, \dots, k\}$ and $\varrho(x) = 1$ otherwise. Then

$$\sum_{g \in \text{supp}(\omega)} \omega(g) \cdot \varrho(g(x_1, \dots, x_k)) = \sum_{g \in \text{Pol}_1(\Gamma)} \omega(g) \cdot \varrho(g(x_1, \dots, x_k)) \leq 0,$$

hence for each $g \in \text{supp}(\omega)$ we have that $\varrho(g(x_1, \dots, x_k)) = 0$, so $g(x_1, \dots, x_k) \in \{x_1, \dots, x_k\}$. This implies that the positive clone of a conservative language is idempotent, and hence every conservative language is a core.

We show that our conjecture agrees with the above complexity classification:

Proposition 42. *Let Γ be a conservative constraint language over a domain D . Then $\text{Pol}^+(\Gamma)$ has an idempotent cyclic operation if and only if Γ admits a conservative binary multimorphism $\langle \sqcap, \sqcup \rangle$ and a conservative ternary multimorphism $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$, and there is a family M of two-element subsets of D , such that:*

- for every $\{x, y\} \in M$, $\langle \sqcap, \sqcup \rangle$ restricted to $\{x, y\}$ is an STP,
- for every $\{x, y\} \notin M$, $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$ restricted to $\{x, y\}$ is an MJN.

Proof. Let Γ' be the language Γ together with all $\{0, \infty\}$ -valued unary cost functions on D . For every weighted polymorphism $\omega \in \text{wPol}(\Gamma)$ there is a corresponding weighted polymorphism of Γ' , which is equal to ω on the conservative operations. Therefore $\text{Pol}^+(\Gamma') = \text{Pol}^+(\Gamma)$, and Γ admits a conservative multimorphism $\langle f_1, \dots, f_k \rangle$ if and only if Γ' does. Now let $\varrho : D \rightarrow \overline{\mathbb{Q}}$ be any general-valued unary cost function. Observe that $\varrho \in \text{wRelClo}(\Gamma')$. It follows that without loss of generality we can assume that Γ contains all general-valued unary cost functions. We do so in the rest of the proof. Observe that every polymorphism of such language is conservative.

Assume that Γ admits the two conservative multimorphisms $\langle \sqcap, \sqcup \rangle$ and $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$ described in the statement of the proposition. There are only four idempotent operations on a two-element domain $\{x, y\}$, namely: \max , \min , π_1 and π_2 (we assume that $\{x, y\} = \{0, 1\}$). Each of the operations \sqcap, \sqcup restricted to any two-element subset $\{x, y\}$ of D must be equal to one of those four. Therefore it is not difficult to prove, using $\{0, 1\}$ -valued unary cost functions, that for every two-element subset $\{x, y\}$ of D :

- either $\langle \sqcap, \sqcup \rangle$ restricted to $\{x, y\}$ is an STP,
- or \sqcap restricted to $\{x, y\}$ is equal to π_1 and \sqcup restricted to $\{x, y\}$ is equal to π_2 (possibly the other way round).

Let M' be the set of those two-element subsets $\{x, y\}$ of D , for which $\langle \sqcap, \sqcup \rangle$ restricted to $\{x, y\}$ is an STP. Obviously $M \subseteq M'$.

Let $t(x, y, z) = ((x \sqcap y) \sqcup (x \sqcap z)) \sqcup (z \sqcap y)$. For every $\{x, y\} \in M'$ we have that t restricted to $\{x, y\}$ is the majority operation. For every other two-element

subset $\{x, y\}$ of D the operation t restricted to $\{x, y\}$ is equal to π_2 or π_3 . Now let us define m to be

$$m(x, y, z) = \text{Mj}_1(t(x, y, z), t(y, z, x), t(z, x, y)).$$

Since the operation Mj_1 is idempotent, for every $\{x, y\} \in M'$ the operation m restricted to $\{x, y\}$ is the majority operation. Moreover, if $\{x, y\}$ does not belong to M' then m restricted to $\{x, y\}$ is equal to Mj_1 (with permuted arguments). But Mj_1 restricted to $\{x, y\}$ is the majority operation. Therefore, m is a majority operation on the whole domain D . If there is a majority operation in the idempotent clone $\text{Pol}^+(\Gamma)$ then there is also an idempotent cyclic operation. This finishes the proof of the right-to-left implication.

The proof of the other implication consists of a sequence of claims and heavily relies on the results of [19].

Assume that $\text{Pol}^+(\Gamma)$ has an idempotent cyclic operation. Let M be a set of all two-element subsets $\{x, y\}$ of D for which there exists no binary cost function $\varrho \in \text{wRelClo}(\Gamma)$ such that

$$(x, y), (y, x) \in \text{Feas}(\varrho), \text{ and } \varrho(x, x) + \varrho(y, y) > \varrho(x, y) + \varrho(y, x).$$

We prove that Γ admits a conservative binary multimorphism $\langle \sqcap, \sqcup \rangle$ and a conservative ternary multimorphism $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$ such that:

- for every $\{x, y\} \in M$, $\langle \sqcap, \sqcup \rangle$ restricted to $\{x, y\}$ is an STP,
- for every $\{x, y\} \notin M$, $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$ restricted to $\{x, y\}$ is an MJN.

Consider the weighted algebra $(D, \text{wPol}(\Gamma))$. Observe that every two-element subset $\{x, y\} \subseteq D$ is a subuniverse of D and let \mathbf{B} be the subalgebra with universe $B = \{x, y\}$.

Claim 1. Every two-element weighted subalgebra \mathbf{B} of $(D, \text{wPol}(\Gamma))$ contains the weighting $\langle \min, \max \rangle$ or $\langle \text{Mjrty}, \text{Mjrty}, \text{Mnrty} \rangle$ (we assume that $B = \{0, 1\}$).

Proof. The weighted algebra $(D, \text{wPol}(\Gamma))$ contains a weighting which assigns a positive weight to an idempotent cyclic operation. Therefore its weighted subalgebra \mathbf{B} contains a weighting which assigns a positive weight to at least one operation that is not a projection, and hence it contains one of the nine weightings listed in Theorem 37. The first three of them are not idempotent. Moreover, for each of the weightings: $\langle \min, \min \rangle$, $\langle \max, \max \rangle$, $\langle \text{Mjrty}, \text{Mjrty}, \text{Mjrty} \rangle$, $\langle \text{Mnrty}, \text{Mnrty}, \text{Mnrty} \rangle$ it is easy to find a $\{0, 1\}$ -valued unary cost function that is not improved by it. We conclude that \mathbf{B} contains the weighting $\langle \min, \max \rangle$ or $\langle \text{Mjrty}, \text{Mjrty}, \text{Mnrty} \rangle$, which finishes the proof of Claim 1.

Claim 2. Let $\{x, y\}$ be a two-element subset of D . There exists no binary cost function $\varrho \in \text{wRelClo}(\Gamma)$ such that

$$(x, y), (y, x) \in \text{Feas}(\varrho), \text{ and } \varrho(x, x) + \varrho(y, y) > \varrho(x, y) + \varrho(y, x),$$

and at least one of the pairs (x, x) , (y, y) belong to $\text{Feas}(\varrho)$.

Proof. Consider the weighted subalgebra \mathbf{B} of $(D, \text{wPol}(\Gamma))$ with the universe $B = \{x, y\}$. Assume that $\{x, y\} = \{0, 1\}$. By Claim 1 the weighted subalgebra \mathbf{B} contains the weighting $\langle \min, \max \rangle$ or $\langle \text{Mjrty}, \text{Mjrty}, \text{Mnrty} \rangle$.

Suppose that \mathbf{B} contains the weighting $\langle \text{Mjrty}, \text{Mjrty}, \text{Mnrty} \rangle$ and let ω be the weighted polymorphisms of Γ that induces $\langle \text{Mjrty}, \text{Mjrty}, \text{Mnrty} \rangle$. Let ϱ be a binary cost function like in the statement of the claim. Without loss of generality assume that $(x, x) \in \text{Feas}(\varrho)$. Then

$$\begin{aligned} \sum_{g \in \text{Pol}_3(\Gamma)} \omega(g) \cdot \varrho(g((x, y), (y, x), (x, x))) &= \frac{1}{3} (2\varrho(\text{Mjrty}((x, y), (y, x), (x, x))) + \\ &+ \varrho(\text{Mnrty}((x, y), (y, x), (x, x)))) - \varrho(x, y) - \varrho(y, x) - \varrho(x, x) = \\ &= \frac{1}{3} (2\varrho(x, x) + \varrho(y, y) - \varrho(x, y) - \varrho(y, x) - \varrho(x, x)) = \\ &= \frac{1}{3} (\varrho(x, x) + \varrho(y, y) - \varrho(x, y) - \varrho(y, x)) > 0. \end{aligned}$$

It follows that ϱ is not improved by ω , so $\varrho \notin \text{wRelClo}(\Gamma)$. Similarly we show that if \mathbf{B} contains the weighting $\langle \min, \max \rangle$ and ω is the weighted polymorphisms of Γ that induces $\langle \min, \max \rangle$, then ϱ is not improved by ω .

Claim 2 together with Theorem 9 of [19] implies that Γ admits a conservative binary multimorphism $\langle \sqcap, \sqcup \rangle$ such that:

- for every $\{x, y\} \in M$, $\langle \sqcap, \sqcup \rangle$ restricted to $\{x, y\}$ is an STP,
- if $\{x, y\} \notin M$ then \sqcap restricted to $\{x, y\}$ is equal to π_1 and \sqcup restricted to $\{x, y\}$ is equal to π_2 .

Claim 3. There exists an operation m in $\text{Pol}^+(\Gamma)$ which is a majority operation.

Proof. First we prove that there exists an operation Mj in $\text{Pol}^+(\Gamma)$ such that for every $\{x, y\} \notin M$ the operation Mj restricted to $\{x, y\}$ is the majority operation. To this end, take any $\{x, y\} \notin M$. By the definition of M there exists a binary cost function $\varrho \in \text{wRelClo}(\Gamma)$ such that

$$(x, y), (y, x) \in \text{Feas}(\varrho), \text{ and } \varrho(x, x) + \varrho(y, y) > \varrho(x, y) + \varrho(y, x).$$

Moreover, by Claim 2 none of the pairs (x, x) , (y, y) belongs to $\text{Feas}(\varrho)$. Therefore:

1. there is no operation in $\text{Pol}^+(\Gamma)$ that restricted to $\{x, y\}$ is the max or min operation (such an operation would not even be a polymorphism of ϱ), and
2. it follows from Claim 1 that there exists an operation f in $\text{Pol}^+(\Gamma)$ such that f restricted to $\{x, y\}$ is the majority operation.

By Proposition 3.1 of [6] it follows from the conditions 1 and 2 above that there exists an operation Mj in $\text{Pol}^+(\Gamma)$ such that for every $\{x, y\} \notin M$ the operation Mj restricted to $\{x, y\}$ is the majority operation.

Let $t(x, y, z) = ((x \sqcap y) \sqcup (x \sqcap z)) \sqcup (z \sqcap y)$. For every $\{x, y\} \in M$ we have that t restricted to $\{x, y\}$ is the majority operation. For every other two-element subset $\{x, y\}$ of D the operation t restricted to $\{x, y\}$ is equal to π_3 . Now let us define m to be

$$m(x, y, z) = \text{Mj}(t(x, y, z), t(y, z, x), t(z, x, y)).$$

Since Mj is idempotent, for every $\{x, y\} \in M$ the operation m restricted to $\{x, y\}$ is the majority operation. Moreover, if $\{x, y\}$ does not belong to M then m restricted to $\{x, y\}$ is equal to Mj (with permuted arguments). But Mj restricted to $\{x, y\}$ is the majority operation. Therefore, m is a majority operation on the whole domain D .

By [19] it follows from Claims 2 and 3 that Γ admits a conservative ternary multimorphism $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$ such that for every $\{x, y\} \notin M$, $\langle \text{Mj}_1, \text{Mj}_2, \text{Mn}_3 \rangle$ restricted to $\{x, y\}$ is an MJN, which finishes the proof of Proposition 42.

5.4 Infinite Constraint Languages.

We finish this section by showing that a variation of Theorem 30 holds for infinite constraint languages. For that, we need to introduce *fractional polymorphisms* which correspond to weighted polymorphisms that can take real values.

An m -ary *fractional operation* ω on D is a probability distribution on $\mathcal{O}_D^{(m)}$. As for weightings, the set of operations to which a fractional operation ω assigns a positive probability is called the *support* of ω and denoted $\text{supp}(\omega)$.

Definition 43. *An m -ary fractional operation ω on D is a fractional polymorphism of a cost function ϱ if, for any list of r -tuples $\mathbf{x}_1, \dots, \mathbf{x}_m \in \text{Feas}(\varrho)$, we have*

$$\sum_{g \in \text{supp}(\omega)} \omega(g) \varrho(g(\mathbf{x}_1, \dots, \mathbf{x}_m)) \leq \frac{1}{m} (\varrho(\mathbf{x}_1) + \dots + \varrho(\mathbf{x}_m)).$$

For a constraint language Γ we denote by $\text{fPol}(\Gamma)$ the set of those fractional operations that are fractional polymorphisms of all cost functions $\varrho \in \Gamma$. Let $\text{fPol}^+(\Gamma) = \{g \in \mathcal{O}_D \mid g \in \text{supp}(\omega), \omega \in \text{fPol}(\Gamma)\}$. It is easy to see that $\text{fPol}^+(\Gamma)$ is a clone (the proof is similar to that of Proposition 12).

Proposition 44. *Let Γ be a finite constraint language. Then $\text{fPol}^+(\Gamma) = \text{Pol}^+(\Gamma)$.*

Proof. Obviously $\text{Pol}^+(\Gamma) \subseteq \text{fPol}^+(\Gamma)$. Let $f \in \text{fPol}^+(\Gamma)$. This can be equivalently expressed by saying that some LP with rational coefficients has a solution. It is well known that every LP with rational coefficients has an optimal solution with rational coefficients (see e.g. [22]). It follows that $f \in \text{Pol}^+(\Gamma)$.

Proposition 45. *There exists an infinite valued constraint language Γ such that $\text{Pol}^+(\Gamma) \subsetneq \text{fPol}^+(\Gamma)$.*

An example of such language is given in [25].

To deal with infinite languages we slightly modify the notion of a core. We say that a valued constraint language Γ is a *core* if all operations in $\text{fPol}_1^+(\Gamma)$ are bijective. It follows from Proposition 44 that for finite languages this definition coincides with the old one. Moreover, the following proposition states that all positively weighted unary polymorphisms can be captured in a finite part of Γ .

Proposition 46. *If a valued constraint language Γ is a core then there exists a finite language $\Gamma' \subseteq \Gamma$ such that $\text{fPol}_1^+(\Gamma) = \text{fPol}_1^+(\Gamma')$.*

We will use the following lemma (Lemma 7 from [25]), which is an immediate consequence of the fact that the space of m -ary fractional operations on a fixed domain D is compact.

Lemma 47. *Let Γ be a valued constraint language and let $O \subseteq \mathcal{O}_D^{(m)}$. If for every finite $\Gamma' \subseteq \Gamma$ there exists a fractional polymorphism with support in O , then Γ has a fractional polymorphism with support in O .*

Proof. (of Proposition 46) Let Γ be a core valued constraint language and let B be the set of such bijective operations f on D that, for all cost functions $\varrho \in \Gamma$, satisfy $\varrho \circ f = \varrho$. A proof analogous to the one of Proposition 16 shows that a unary fractional operation ω is a fractional polymorphism of Γ if and only if $\text{supp}(\omega) \subseteq B$.

Take a language Γ' such that $\text{fPol}_1^+(\Gamma') \neq \text{fPol}_1^+(\Gamma)$. This means that there exists a fractional polymorphism ω of Γ' such that there is some operation g from outside B in $\text{supp}(\omega)$. Then there exists a fractional polymorphism ω' of Γ' such that $\text{supp}(\omega') \subseteq \mathcal{O}_D^{(1)} \setminus B$. We construct it from the fractional polymorphism ω by assigning probability 0 to all operations in B and scaling to get a proper probability distribution. It is not difficult to check that the obtained fractional operation ω' is a fractional polymorphism of Γ' .

Now suppose that none of the finite $\Gamma' \subseteq \Gamma$ satisfies $\text{fPol}_1^+(\Gamma) = \text{fPol}_1^+(\Gamma')$. Then every finite $\Gamma' \subseteq \Gamma$ has a fractional polymorphism with support in $\mathcal{O}_D^{(1)} \setminus B$. It follows from Lemma 47 that Γ has a fractional polymorphism with support in $\mathcal{O}_D^{(1)} \setminus B$, which is a contradiction.

A proof analogous to the one of Proposition 14 shows that every constraint language Γ has a computationally equivalent language Γ' which is a core (in the new sense).

Proposition 48. *For every valued constraint language Γ there exists a core language Γ' , such that the valued constraint language Γ is tractable if and only if Γ' is tractable, and it is NP-hard if and only if Γ' is NP-hard.*

Moreover, using Proposition 46, we can show that constants can be added to a core language (the proposition implies that, for a core language Γ , the relation characterizing $\text{fPol}_1^+(\Gamma)$ belongs to $\text{wRelClo}(\Gamma)$). The proof follows along the same lines as that of Proposition 20.

Proposition 49. *Let Γ be a valued constraint language which is a core. The valued constraint language Γ_c is a rigid core. Moreover, Γ is tractable if and only if Γ_c is tractable, and Γ is NP-hard if and only if Γ_c is NP-hard.*

Theorem 50. *Let Γ be a core valued constraint language. If, for every finite $\Gamma' \subseteq \Gamma$, the set $\text{fPol}^+(\Gamma')$ has a Taylor operation, then so does $\text{fPol}^+(\Gamma)$.*

Proof. Consider a finite $\Gamma' \subseteq \Gamma$. Since Γ is a core, by Proposition 46 there exists a finite $\hat{\Gamma} \subseteq \Gamma$ which is a core. By $\overline{\Gamma'}$ we denote the language $\Gamma' \cup \hat{\Gamma}$. It is a finite subset of Γ , and hence it has a Taylor operation. Fix some prime number $p > |D|$. It follows from Theorem 30 and Theorem 33 that $\text{Pol}^+(\overline{\Gamma'})$ has an idempotent cyclic operation of arity p , and therefore by Proposition 39 $\overline{\Gamma'}$ admits an idempotent cyclic weighted polymorphism ω' of arity p .

Let O denote the set of p -ary idempotent cyclic operations on D . From the weighted polymorphism ω' it is easy to construct a fractional polymorphism $\overline{\omega'}$ such that $\text{supp}(\overline{\omega'}) \subseteq O$. Since $\Gamma' \subseteq \overline{\Gamma'}$, we have that $\overline{\omega'}$ is a fractional polymorphism of Γ' with support in O . This holds for every finite $\Gamma' \subseteq \Gamma$. Hence, by Lemma 47 there exists a fractional polymorphism ω of Γ with support in O .

It immediately follows that if Γ does not have a Taylor operation in $\text{fPol}^+(\Gamma)$ then it is NP-hard.

Corollary 51. *Let Γ be a core valued constraint language. If $\text{fPol}^+(\Gamma)$ does not have a Taylor operation, then Γ is NP-hard.*

References

1. Libor Barto and Marcin Kozik. Absorbing subalgebras, cyclic terms, and the constraint satisfaction problem. *Logical Methods in Computer Science*, 8(1), 2012.
2. G. Birkhoff. On the structure of abstract algebras. In *Proceedings of the Cambridge Philosophical Society*, volume 31, pages 433–454, 1935.
3. Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34:720–742, 2005.
4. Andrei A. Bulatov. Tractable conservative constraint satisfaction problems. In *Proc. of the 18th Symposium on Logic in Computer Science*, page 321, 2003.
5. Andrei A. Bulatov. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM*, 53(1):66–120, 2006.
6. Andrei A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Logic*, 12(4):24:1–24:66, 2011.
7. Andrei A. Bulatov, Andrei A. Krokhin, and Peter Jeavons. Constraint satisfaction problems and finite algebras. In *Proc. ICALP '00*, pages 272–282, 2000.
8. S. Burris and H.P. Sankappanavar. *A course in universal algebra*. Graduate texts in mathematics. Springer-Verlag, 1981.
9. David A. Cohen, Martin C. Cooper, Páidí Creed, Peter G. Jeavons, and Stanislav Živný. An algebraic theory of complexity for discrete optimization. *SIAM J. Comput.*, 42(5):1915–1939, 2013.
10. David A. Cohen, Martin C. Cooper, Peter G. Jeavons, and Andrei A. Krokhin. The complexity of soft constraint satisfaction. *Artif. Intell.*, 170(11):983–1016, 2006.

11. J. Farkas. Über die theorie der einfachen ungleichungen. *J. Reine Angewandte Math.*, 124:1–24, 1902.
12. Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1999.
13. P. Gordan. Über die auflösungen linearer gleichungen mit reelen coefficienten. *Mathematische Annalen*, 6:23–28, 1873.
14. Pavol Hell and Jaroslav Nešetřil. On the complexity of h-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92 – 110, 1990.
15. Anna Huber, Andrei Krokhin, and Robert Powell. Skew bisubmodularity and valued csps. In *Proc. SODA '13*, pages 1296–1305. SIAM, 2013.
16. Peter Jeavons, David Cohen, and Marc Gyssens. Closure properties of constraints. *J. ACM*, 44(4):527–548, 1997.
17. Peter Jeavons, Andrei Krokhin, and Stanislav Živný. The complexity of valued constraint satisfaction. *Bulletin of the EATCS*, 113:21–55, 2014.
18. Vladimir Kolmogorov, Johan Thapper, and Stanislav Živný. The power of linear programming for general-valued csps. *CoRR*, abs/1311.4219, 2013.
19. Vladimir Kolmogorov and Stanislav Živný. The complexity of conservative valued csps. *J. ACM*, 60(2):10:1–10:38, 2013.
20. Benot Larose and Pascal Tesson. Universal algebra and hardness results for constraint satisfaction problems. *Theor. Comput. Sci.*, 410:1629 – 1647, 2009.
21. Thomas J. Schaefer. The complexity of satisfiability problems. In *Proc. of the 10th ACM Symp. on Theory of Computing*, STOC '78, pages 216–226, 1978.
22. Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.
23. Alfred Tarski. A remark on functionally free algebras. *Annals of Mathematics*, 47(1):163–166, 1946.
24. Walter Taylor. Varieties obeying homotopy laws. *Canad. J. Math.*, 29(3):498527, 1997.
25. Johan Thapper and Stanislav Živný. The complexity of finite-valued csps. In *Proc. of the 45th ACM Symp. on Theory of Computing*, STOC '13, pages 695–704, 2013.